



National Technical University of Athens
School of Mechanical Engineering
Fluids Department
Laboratory of Thermal Turbomachines
Parallel CFD & Optimization Unit

Programming of a Mesh Displacement Method using Radial Basis Functions and Quaternions

Diploma Thesis
by

Vasileios Georgopoulos

Advisor:
Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2021

Acknowledgements

At this stage I feel the need to thank all those who contributed to the elaboration of this diploma Thesis.

First of all, I would like to thank my professor K.C. Giannakoglou, for the opportunity he gave me to deal with such an interesting topic. In addition, I would like to thank him for the valuable guidance he has provided me whenever I needed it as well as for the patience he has shown throughout the elaboration of this diploma Thesis. Its completion would in no way be possible without his contribution.

I would also like to express my special thanks to Dr. V. Asouti and Dr. E. Papoutsis as they have always been willing to help me in whatever I needed.

I could not miss my parents, Dimitris and Ioanna. I am grateful to them for their support throughout all these years. They were always by my side supporting every choice of me. Finally, I want to thank my friends for their understanding and for the fact that they have always been there.



National Technical University of Athens
School of Mechanical Engineering
Fluids Department
Parallel CFD & Optimization Unit

Programming of a Mesh Deformation Method using Radial Basis Functions and Quaternions

Diploma Thesis

Vasileios Georgopoulos

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2021

Abstract

The need for adapting an existing mesh to displaced boundaries arises in many applications of Computational Fluid Dynamics. In aerodynamic shape optimization problems, this is necessary either if a shape parameterization tool is used, in which case the boundaries of the computational mesh are modified in each cycle of the optimization loop and a mechanism propagating the surface displacements to the interior mesh nodes must be devised, or in parameterization-free methods, where the same mechanism undertakes the displacement of internal and boundary nodes. Furthermore, in aeroelasticity problems with fluid-structure interaction, structure's shape changes at each time-step as the solid body, such as a wing, oscillates. The mesh must be readjusted to conform to the new geometry. Another instance where a mesh deformation tool is needed is in flow simulations in the presence of moving bodies since the mesh should be deformed accordingly every time one or more solid parts move. In all the above cases, remeshing is extremely costly for industrial-scale applications, and it might also hinder the continuation of new simulations from available numerical solutions on the unstructured mesh, for instance, of the previous domain.

As mentioned above, remeshing is highly costly for large-scale applications. It may also lead to interpolation errors from the old to the new mesh. Therefore, an automated algorithm that morphs the existing mesh according to updated solid geometry is used. The development of such a mesh deformation algorithm is the main subject of this diploma thesis. On this account, the proposed method synergistically employs a radial basis function (RBF) network and quaternions. RBF networks use radial basis functions as activation functions and are extensively em-

ployed in mesh morphing applications. Quaternions are four-element vectors that are ideal for modeling rotations in three-dimensional space. This thesis proposes a hybrid approach that couples the standard RBF method with quaternions in order to produce a higher quality mesh near the solid boundary, where greater gradients of pressure and velocity occur. Metrics regarding the quality of unstructured grids are also presented and used in order to quantitatively assess the deformed meshes.



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής
& Βελτιστοποίησης

Προγραμματισμός Μεθόδου Παραμόρφωσης Πλέγματος με Συναρτήσεις Ακτινικής Βάσης και Τετραδόνια

Διπλωματική εργασία

Βασίλειος Γεωργόπουλος

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2021

Η ανάγκη προσαρμογής ενός υφιστάμενου μη-δομημένου πλέγματος σε μετατοπισμένα όρια προκύπτει σε πολλές εφαρμογές της Υπολογιστικής Ρευστοδυναμικής. Σε προβλήματα αεροδυναμικής βελτιστοποίησης, αυτό είναι απαραίτητο είτε στην παραλλαγή όπου χρησιμοποιείται παραμετροποίηση της προς σχεδιασμό μορφής, η οποία τροποποιεί σε κάθε κύκλο της βελτιστοποίησης τα όρια του υπολογιστικού χωρίου και αυτό που χρειάζεται είναι να προσαρμοστεί κατάλληλα το εσωτερικό πλέγμα, είτε στην παραλλαγή που ενιαίος μηχανισμός αναλαμβάνει, ελλείψει παραμετροποίησης, την ταυτόχρονη και συμβατή μετακίνηση όλων των κόμβων ενός υπολογιστικού πλέγματος, συμπεριλαμβανομένων των ορίων. Επιπλέον, σε προβλήματα αεροελαστικότητας με την αλληλεπίδραση ρευστού-στερεού, το σχήμα του στερεού αλλάζει σε κάθε χρονικό βήμα καθώς το στερεό σώμα, όπως μία πτέρυγα, ταλαντώνεται και το πλέγμα πρέπει να ρυθμιστεί εκ νέου για να συμμορφωθεί με τη νέα γεωμετρία. Μια άλλη περίπτωση όπου απαιτείται εργαλείο παραμόρφωσης πλέγματος είναι σε προσομοιώσεις ροής παρουσία κινούμενων σωμάτων, αφού το πλέγμα πρέπει να παραμορφώνεται ανάλογα κάθε φορά που ένα ή περισσότερα στερεά μέρη μετακινούνται. Σε όλες τις παραπάνω περιπτώσεις, η επαναπλεγματοποίηση είναι εξαιρετικά δαπανηρή για εφαρμογές βιομηχανικής κλίμακας και μπορεί επίσης να εμποδίσει τη συνέχιση νέων προσομοιώσεων από διαθέσιμες αριθμητικές λύσεις στο μη-δομημένο πλέγμα, για παράδειγμα, του προηγούμενου χωρίου.

Όπως αναφέρθηκε παραπάνω, η γένεση πλέγματος είναι εξαιρετικά δαπανηρή για εφαρμογές μεγάλης κλίμακας και μπορεί επίσης να οδηγήσει σε σφάλματα παρεμβολής από το παλιό στο νέο πλέγμα. Ως εκ τούτου, χρησιμοποιείται ένας αυτοματοποιημένος αλγόριθμος που διαμορφώνει το υπάρχον πλέγμα σύμφωνα με τη νέα γεωμετρία. Η ανάπτυξη ενός τέτοιου αλγορίθμου παραμόρφωσης πλέγματος είναι το κύριο αντικείμενο αυτής της διπλωματικής εργασίας. Για το λόγο αυτό, η προτεινόμενη μέθοδος χρησι-

μποιεί δίκτυα ακτινικής βάσης (RBF) και τετραδόνια. Τα δίκτυα (RBF) χρησιμοποιούν συναρτήσεις ακτινικής βάσης ως συναρτήσεις ενεργοποίησης και χρησιμοποιούνται ευρέως σε εφαρμογές μορφοποίησης πλέγματος. Τα τετραδόνια είναι διανύσματα τεσσάρων στοιχείων που είναι ιδανικά για μοντελοποίηση περιστροφών στον τριδιάστατο χώρο. Η διπλωματική αυτή εργασία προτείνει μια υβριδική προσέγγιση που συνδυάζει τη γνωστή μέθοδο (RBF) με τετραδόνια για να παράγει πλέγμα υψηλότερης ποιότητας κοντά στο στερεό όριο, όπου εμφανίζονται μεγαλύτερες κλίσεις πίεσης και ταχύτητας. Επίσης, παρουσιάζονται και χρησιμοποιούνται μετρικές σχετικά με την ποιότητα των μη-δομημένων πλεγμάτων, προκειμένου να εκτιμηθεί ποιοτικά το παραμορφωμένο πλέγμα.

Acronyms

CFD	Computational Fluid Dynamics
NTUA	National Technical University of Athens
PCOpt	Parallel CFD & Optimization unit
RBF	Radial Basis Function
CPU	Central Processing Unit
PDE	Partial Differential Equation
RHS	Right Hand Side
w.r.t.	with respect to

ΕΜΠ	Εθνικό Μετσόβιο Πολυτεχνείο
ΕΘΣ	Εργαστήριο Θερμικών Στροβιλομηχανών
ΜΠΤΡ&Β	Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης
ΥΡΔ	Υπολογιστική Ρευστοδυναμική
ΜΔΕ	Μερική Διαφορική εξίσωση

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Other mesh deformation methods	2
1.3	Thesis outline	3
2	Radial Basis Functions Networks	5
2.1	RBF-Based Interpolation	5
2.2	Radial Basis Functions	6
2.2.1	Globally supported radial basis functions	6
2.2.2	Compactly supported radial basis functions	10
2.3	RBF Mesh Morphing	12
3	Quaternions	15
3.1	Introduction	15
3.2	Quaternions Background	16
3.2.1	Quaternion Arithmetic	17
3.3	Quaternions and spatial rotation	19
4	The proposed hybrid approach	23
4.1	Construction of quaternions on boundaries	23
4.2	Method description	26
5	Application to the mesh around a rotated sycamore leaf	29

5.1	Problem description	29
5.2	Results	31
5.2.1	Inverse quadratic	31
5.2.2	Inverse multi-quadric	34
5.2.3	Wendland C^0 ($R_s = 2$)	36
5.2.4	Wendland C^0 ($R_s = 5$)	38
5.2.5	Wendland C^0 ($R_s = 11$)	40
5.2.6	Wendland C^2 ($R_s = 11$)	42
5.2.7	Wendland C^2 ($R_s = 11$) with Gaussian ($\varepsilon = 0.5$) as weighting function	44
5.2.8	Wendland C^2 ($R_s = 11$) with Gaussian ($\varepsilon = 1$) as weighting function	46
5.2.9	Wendland C^2 ($R_s = 11$) with Wendland C^0 as weighting func- tion	48
5.3	Comments	50
6	Application to the mesh around a rotated four element airfoil	51
6.1	Problem description	51
6.2	Results	53
6.2.1	Exponential	53
6.2.2	Wendland C^0	56
6.2.3	Wendland C^2	58
6.2.4	Wendland C^2 ($R_s = 2$) with Gaussian as weighting function .	60
6.2.5	Wendland C^2 ($R_s = 2$) with Wendland C^0 as weighting function	63
6.3	Comments	64
7	Application to the mesh around a four element airfoil with moving slats and flaps	67
7.1	Problem description	67
7.2	Results	68

7.2.1	Exponential	68
7.2.2	Wendland C^0	73
7.2.3	Wendland C^2	77
7.2.4	Wendland C^0 ($R_s = 5$) with Exponential weighting function	81
8	Application to the mesh around an aircraft	83
8.1	Problem description	83
8.2	Results	86
8.2.1	Wendland C^0	86
8.2.2	Wendland C^2	88
8.2.3	Wendland C^2 ($R_s = 40$) with Gaussian as weighting function	92
9	Aggregate results	95
9.1	Rotated sycamore Leaf	95
9.2	Rotated four-element airfoil	96
9.3	Aircraft	97
10	Summary - Conclusion	99
A	Measuring mesh quality	101
A.1	Impact of mesh quality	101
A.2	Mesh quality metrics	102
A.2.1	Triangle relative size metric	102
A.2.2	Triangle shape metric	103

Chapter 1

Introduction

1.1 Motivation

The advent of computers has allowed the possibility of extensive numerical simulations of real physical problems. With engineering design trends towards more complicated structures, computational modeling is currently gaining importance. Numerical simulations become essentially convenient when experiments become difficult, dangerous, or are simply too expensive to perform. Additionally, in many cases, it has the advantage of providing detailed results that are much easier for humans to process. Computational fluid dynamics (CFD) is a branch of fluid mechanics that analyzes and solves such problems, which involve fluid flows. To achieve that, the continuous geometry space where simulation occurs should be discretized through a mesh that divides a geometry into many elements. There are many cases where the structure to be studied has moving boundaries or undergoing deformations. In such problems, the generated or deformed mesh is crucial to conform to the new geometry of the structure. The generation of a new mesh every time the structure's geometry changes is computationally expensive. Thus, a mesh deformation technique is used as a much more feasible alternative solution.

Aerodynamic shape optimization consists in maximizing the performance of a given body by changing its shape. The performance is evaluated using a CFD solver and the optimization process is iterative: it starts with a given shape and then changes that shape in every optimization cycle to achieve improved performance. Thus, the mesh must be adapted to this new geometry in order to be used by the CFD solver in the following optimization cycle.

Furthermore, in aeroelasticity problems with fluid-structure interaction (FSI), structure's shape changes at each time step (e.g., the shape of a wind turbine's wing

changes due to the time-varying wind force applied on it). In such cases, these deformations are often caused from the oscillations of the solid body. Similar to the aerodynamic optimization process, to solve the flow at the new time step, the mesh must be readjusted to conform to the deformed geometry of the object.

Another instance where a mesh deformation tool is needed is when the solid structure contains one or more moving parts (e.g., slats and flaps of a multi-element airfoil). The mesh should be deformed accordingly every time that one or more solid parts move. In the above cases, it is considered that the solid geometry is not excessively deformed at each cycle or time step. If the last is expected for the aerodynamic optimization and aeroelasticity simulations, that would not be the case for a simulation involving a geometry with moving parts. Extensive deformations may lead to a low-quality (i.e., contains irregular elements) or even to an unsuitable mesh (i.e., contains inverted elements).

As mentioned above, remeshing is extremely costly for large-scale applications, and it may also lead to interpolation errors from the old to the new mesh [1]. Therefore, an automated algorithm that morphs the existing mesh according to updated solid geometry is used. The development of such a mesh deformation algorithm is the main subject of this diploma thesis. On this account, the proposed method utilizes a radial basis function (RBF) network and quaternions. RBF networks use radial basis functions as activation functions and are extensively employed in mesh morphing applications [2], [3]. They essentially propagate the known displacements of a set of mesh nodes, typically boundary nodes, into the volume of the mesh. Quaternions are four-element vectors that are ideal for modeling rotations in three-dimensional space. Among others, they also used in mesh deformation techniques [4], [5] representing any known displacement as the composition of a rotation and a translation. This thesis proposes a hybrid approach that couples the known RBF method with quaternions in order to produce a higher quality mesh near the solid boundary, where greater gradients of pressure and velocity occur.

1.2 Other mesh deformation methods

The need to avoid the computationally burdensome re-meshing procedure led to the development of several methods that modulate the existing mesh according to the deformation of the solid geometry. Various methods have been proposed to meet specific requirements emerging from diverse types of simulations or even disciplines. These methods may be classified as interpolations, control meshes and physical analogies [6], algebraic vs. partial differential equation (PDE) methods [7], connectivity-based vs. point-based methods [8], methods for structured or unstructured meshes [9] and methods that can be parallelized [10]. Surveys [6,11] itemize the characteristics of mesh displacement methods according to applications and quality.

Spring analogy methods model the mesh as a network of springs and solve the static

equilibrium equations to find the updated locations of mesh nodes. The springs may be linear [12], torsional [13], semi-torsional [14] or ball-vertex [15]. The above methods are efficient for meshes used in computational structure mechanics (CSM). On the other hand, Selim and Koomullil [6] reported invalid elements and high computational cost when the method dealt with large displacements of CFD meshes with stretched elements for viscous flow simulations.

Continuum elastic methods have been referred in several papers (i.e., [16–18]). They confront the mesh as a solid elastic body, and they displace it by solving the linear elasticity equations. Laplacian methods [19] solve the PDEs to diffuse the known mesh node displacement into the volume. At increased computational cost, solving the bi-harmonic smoothing equation [20] can attain a better quality of the deformed mesh. The algebraic damping method [21] move each internal mesh node in terms of the known displacement of the closest boundary node. The results are satisfactory for small mesh deformations, and the mesh displacement is very rigid near the boundaries but, for larger deformations, algebraic smoothing [22] might be necessary to improve deformed mesh quality.

The Delaunay graph method [23] is based on the generation of a control mesh based on the Delaunay triangulation of boundary nodes and the mapping of the internal nodes on the Delaunay graph. The triangulation adapts to boundary displacements, and the new location of internal mesh nodes is computed using barycentric interpolation. The Inverse Distance Weighting (IDW) method [24] relocates the volume mesh nodes through direct interpolation of the known boundary displacements using weights that depend on the distance of the internal node from the boundary. Transfinite interpolation [25] interpolates the deformations along mesh lines in structured meshes; hence it is pretty computationally effective but incompatible with unstructured grids.

1.3 Thesis outline

The present diploma thesis focuses in the development of a mesh deformation algorithm that uses an RBF network and quaternions defined on boundary nodes.

In Chapter 2, the RBF-based interpolation is presented together with some commonly used radial basis functions. Both of the above are utilized to form the RBF mesh morphing method.

In Chapter 3, quaternions are introduced as a handy way to model rotations.

In Chapter 4, the proposed hybrid method that couples RBF networks with quaternions is described.

In Chapter 5, the developed software is applied to the mesh around a rotated sycamore leaf located within a stationary circle.

In Chapter 6, an application to the mesh around a rotated four-element airfoil is examined.

In Chapter 7, an application to the mesh around the same four-element airfoil, but this time instead of rotating, slats and flaps of the airfoil move upwards and downwards.

In Chapter 8, the developed software is applied to the mesh around a 3D aircraft.

In Chapter 9, some aggregate results are presented regarding the previous applications.

In Chapter 10, a summary of the present diploma thesis is presented.

Chapter 2

Radial Basis Functions Networks

RBF networks use radial basis functions as activation functions. They were first formulated by Broomhead and Lowe in a 1988 paper [26], [27]. The output of an RBF network is a weighted summation of the radial basis functions of the inputs and neuron parameters. Among others, they are extensively used in mesh morphing applications exploiting their ability to interpolate scattered data or multivariate functions in multidimensional space.

2.1 RBF-Based Interpolation

An RBF interpolation at a point \mathbf{x} consists in the weighted linear combination of inter-distance interactions of that point with a given set of source points. The inter-distance interaction is derived from a basis function ϕ with respect to Euclidean distance $\|\cdot\|$.

Let $\Omega \subset \mathbb{R}^Q$ be a Q -dimensional space and a subset $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \Omega$ which contains N data centers (i.e. interpolation sources). Additionally, let $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_N)^T$ be the values of data centers to be interpolated. Assuming $h : \Omega \rightarrow \mathbb{R}$ denotes the kernel interpolation of N data centers, it then takes the following form

$$h(\mathbf{x}) = \sum_{n=1}^N \mathbf{w}_n K(\mathbf{x}, \mathbf{x}_n) \quad (2.1)$$

Here $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_N)^T$, $\mathbf{w}_n \in \mathbb{R}^Q$, $n = 1, \dots, N$ are the so-called interpolation weights and $K : \Omega \times \Omega \rightarrow \mathbb{R}$ is the reproducing kernel of the Hilbert space $\mathcal{H}(K, \Omega)$ equipped with an inner product which allows lengths (i.e. norms) and angles to

be defined. For more details on reconstructing kernel Hilbert space (RKHS), refer to [28].

If there exists a function $\phi : [0, \infty) \rightarrow \mathbb{R}$ such that $K(\mathbf{x}_1, \mathbf{x}_2) = \phi(\|\mathbf{x}_1 - \mathbf{x}_2\|)$ for every $(\mathbf{x}_1, \mathbf{x}_2) \in \Omega \times \Omega$, then K is a radial kernel and ϕ is an RBF kernel. Equation 2.1 then becomes

$$h(\mathbf{x}) = \sum_{n=1}^N \mathbf{w}_n \phi(\|\mathbf{x} - \mathbf{x}_n\|) \quad (2.2)$$

The interpolation weights are determined by demanding that h will produce the exact values of N data centers, i.e.

$$h(\mathbf{x}_n) = \mathbf{y}_n, \quad n = 1, \dots, N \quad (2.3)$$

Equation 2.3 can be written in matrix form as

$$\Phi \mathbf{w} = \mathbf{y} \quad (2.4)$$

where

$$\Phi = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \dots & \phi_N(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_N) & \dots & \phi_N(\mathbf{x}_N) \end{bmatrix} \quad (2.5)$$

and $\phi_i(\mathbf{x}_j) = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$

2.2 Radial Basis Functions

Radial basis functions (RBFs) are real-valued functions depending only on the distance of a point $\mathbf{x} \in \mathbb{R}^Q$ from a given center $\mathbf{x}_n \in \mathbb{R}^Q$, $n = 1, \dots, N$.

There are numerous radial basis functions suitable for interpolating functions and scattered data in Q -dimensional space. The behavior of the interpolation and the extrapolation strongly depends on the chosen radial function [2]. RBFs can be divided into two groups according to the interpolation space they cover: ones with global support and ones with compact support.

2.2.1 Globally supported radial basis functions

RBFs with global support cover the whole interpolation space, so all the cloud points interact with each other. Therefore, using global support achieves high accuracy but leads to dense interpolation matrices, making the numerical problem challenging. In Table 2.1 four globally supported RBFs are given. Parameter ε is the so-called

shape-parameter, and it is used to tune the shape of the RBF.

Name	Abbreviation	$\phi(\mathbf{r})$
Gaussian	GS	$e^{-(\varepsilon r)^2}$
Multiquadric	MQ	$\sqrt{1 + (\varepsilon r)^2}$
Inverse quadratic	IQ	$\frac{1}{1+(\varepsilon r)^2}$
Inverse multiquadric	IMQ	$\frac{1}{\sqrt{1+(\varepsilon r)^2}}$

Table 2.1: *Global supported RBFs*

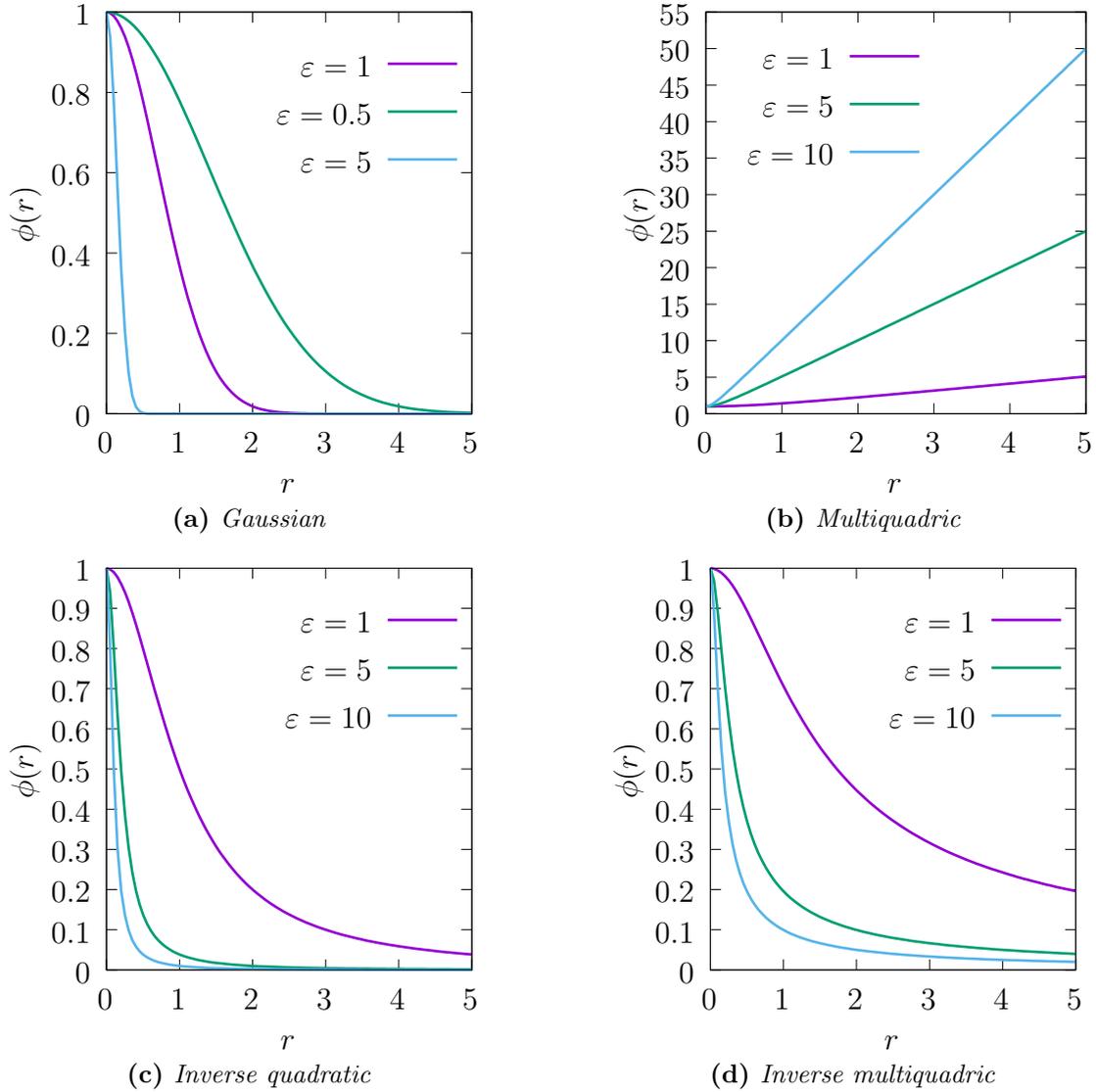


Figure 2.1: Global supported radial basis functions for several choices of shape parameter ε

Shape-parameter tuning

Radial basis functions of Table 2.1 contain a parameter, ε , that controls the shape of the given function. In particular, it regulates their relative flatness or peakedness with the function becoming increasingly flat as $\varepsilon \rightarrow 0$ and vice-versa. As an example, using Gaussian functions, Figure 2.2 illustrates above behavior. A flatter curve essentially means that the function strongly influences a wider area, while a more peaked RBF tends to have a more localized influence. The choice of ε has a profound impact on both the accuracy and numerical stability of the solution to the interpolation problem. As seen in Figure 2.1 every RBF is scaled such that $\phi(0) = 1$.

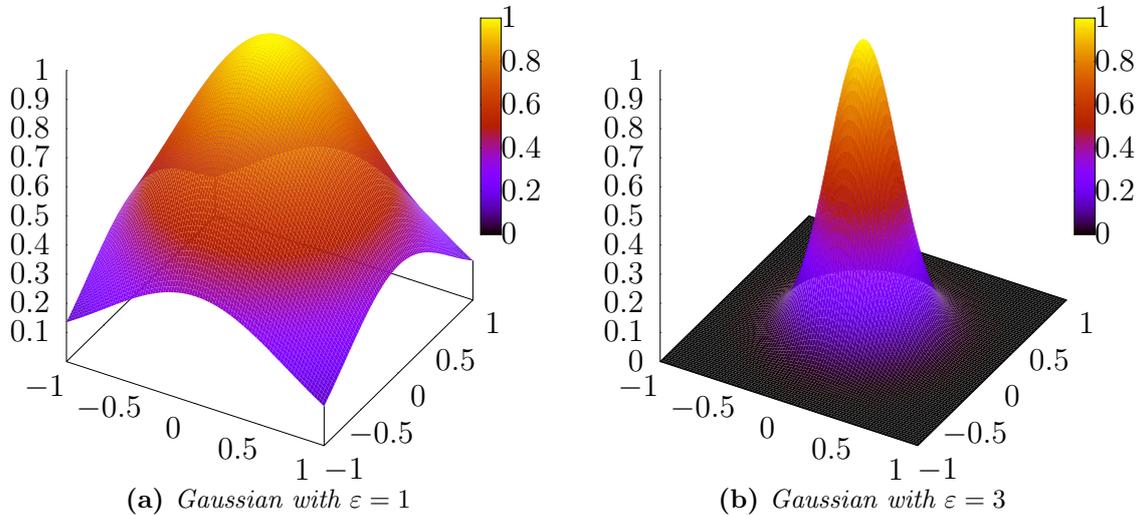


Figure 2.2: Shape-parameter impact on the Gaussian's shape

As a result, the interpolation matrix Φ (Equation 2.5) approaches the identity matrix as $\varepsilon \rightarrow \infty$ leading to matrices with small condition number and thus numerical stability when solving the matrix system. However, the resulting interpolant will in general be a poor approximation of the function, or the scattered data, since it will sharply peak near the interpolation points and, everywhere else, will be near zero. An interpolant with the above behavior is the so-called *bed-of-nails interpolant* and an example of such an interpolant is depicted in Figure 2.3.

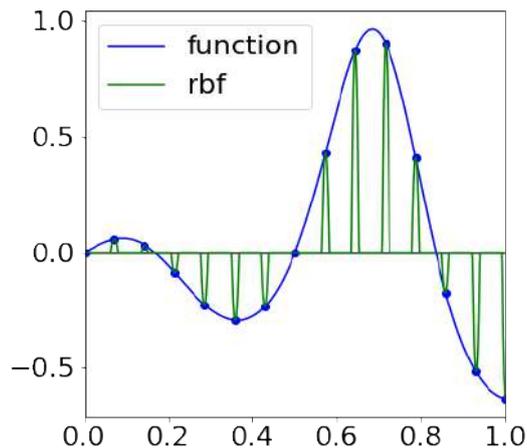


Figure 2.3: An RBF interpolant of the function $f(x) = e^{x \cos 3\pi x} - 1$ sampled at 15 points, using Gaussians, with a shape-parameter $\varepsilon = 100$. The bed-of-nails interpolant [29]

On the other hand, as $\varepsilon \rightarrow 0$, the approximation becomes better, but the condition number of the interpolation matrix Φ will diverge to infinity leading to ill-conditioned systems. From the above, it follows that a trade-off exists between accuracy and numerical stability; hence the choice of the shape-parameter must be such that the interpolation matrix is at the edge of ill-conditioning [30].

2.2.2 Compactly supported radial basis functions

RBF kernels with compact support were introduced in order to reduce the computational cost of the interpolation [2]. They are piecewise-defined functions which satisfy the following property

$$\phi(r, r_s) = \begin{cases} f(r, r_s) & \text{if } r \leq r_s \\ 0 & \text{if } r > r_s \end{cases} \quad (2.6)$$

where $r = \|\mathbf{x} - \mathbf{x}_{sn}\|$, \mathbf{x} is the position vector of a point in space, \mathbf{x}_{sn} the position vector of the n interpolation source point and r_s is the support radius, which determines the region of influence of the kernel around each interpolation source. Let $\xi = \frac{r}{r_s}$, the Equation 2.6 can be written in dimensionless form

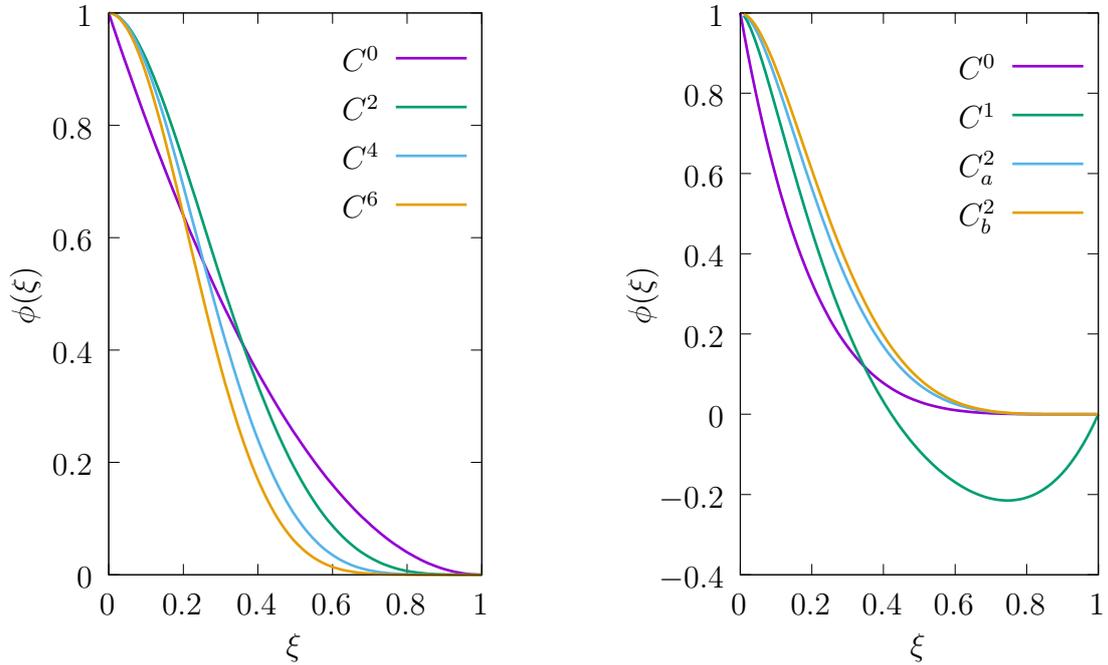
$$\phi(\xi) = \begin{cases} f(\xi) & \text{if } \xi \leq 1 \\ 0 & \text{if } \xi > 1 \end{cases} \quad (2.7)$$

Compactly supported RBFs cover only a specific region which is determined by the radius of influence r_s . Each RBF kernel influences all the points within the cycle (or sphere in 3-dimensional space) of radius r_s . It follows that the smaller the radius of influence is, the less dense the interpolation matrix is. Therefore, RBF kernels with compact support may lead to sparse matrices that can be solved more efficiently. However, a trade-off between the mesh quality and the computational cost exists, meaning that for large deformations, the radius of influence must be increased, and the problem becomes similar to one using globally supported RBFs, vanishing the benefits of compact support [2].

In Table 2.2 some commonly used RBFs with compact support are presented as proposed from Wendland [31]. The first four are based on polynomials that are chosen such that they have the lowest degree of all polynomials that create a C^n continuous basis function with $n \in \{0, 2, 4, 6\}$. The last four functions are based on the thin-plate spline which create C^n continuous basis functions with $n \in \{0, 1, 2\}$. There are two possible CTPS C^2 functions distinguished by subscript a and b .

Name	$\phi(r) = f(\xi), \xi = \frac{r}{r_s} \leq 1$
CP C^0	$(1 - \xi)^2$
CP C^2	$(1 - \xi)^4 (4\xi + 1)$
CP C^4	$(1 - \xi)^6 \left(\frac{35}{3}\xi^2 + 6\xi + 1\right)$
CP C^6	$(1 - \xi)^8 (32\xi^3 + 24\xi^2 + 8\xi + 1)$
CTPS C^0	$(1 - \xi)^5$
CTPS C^1	$1 + \frac{80}{3}\xi^2 - 40\xi^3 + 15\xi^4 - \frac{8}{3}\xi^5 + 20\xi^2 \log(\xi)$
CTPS C_a^2	$1 - 30\xi^2 - 10\xi^3 + 45\xi^4 - 6\xi^5 - 60\xi^3 \log(\xi)$
CTPS C_b^2	$1 - 20\xi^2 + 80\xi^3 - 45\xi^4 - 16\xi^5 + 60\xi^4 \log(\xi)$

Table 2.2: Radial basis functions with compact support



(a) Compact supported RBFs based on polynomials

(b) Compact supported RBFs based on thin plate spline

Figure 2.4: RBFs with compact support as proposed from Wendland [31]

2.3 RBF Mesh Morphing

RBFs are nowadays well accepted as one of the most powerful and versatile mathematical tools in mesh morphing applications. When RBFs are used in such applications, the quantity to be interpolated is the displacement at each node of the mesh, and the interpolation function takes the form

$$\mathbf{d}(\mathbf{x}) = \sum_{n=1}^N \mathbf{w}_n \phi(\|\mathbf{x} - \mathbf{x}_{s_n}\|) \quad (2.8)$$

where $\mathbf{x}_{s_n} \in \mathbb{R}^3$ are the N source nodes in which the displacement values are known. Typically as source points are chosen, all the boundary points, or part of them through a sub-sampling algorithm, and as targets, all the points of the field mesh to be morphed. Alternatively, a sub-sampling algorithm can choose a subset of the boundary points as sources in favor of computational cost. In the training phase, the weights $\mathbf{w}_n \in \mathbb{R}^3$ are computed, so the Equation 2.8 exactly reproduce the known displacements $\boldsymbol{\delta}_n \in \mathbb{R}^3$, $n = 1, \dots, N$ at each source node. This requires solving three $N \times N$ linear systems, one for each spatial direction.

$$\underbrace{\begin{bmatrix} \phi_1(\mathbf{x}_{s_1}) & \dots & \phi_N(\mathbf{x}_{s_1}) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_{s_N}) & \dots & \phi_N(\mathbf{x}_{s_N}) \end{bmatrix}}_{\boldsymbol{\Phi}_s} \underbrace{\begin{bmatrix} \mathbf{w}_1^\top \\ \vdots \\ \mathbf{w}_N^\top \end{bmatrix}}_W = \underbrace{\begin{bmatrix} \boldsymbol{\delta}_1^\top \\ \vdots \\ \boldsymbol{\delta}_N^\top \end{bmatrix}}_{\Delta_s} \quad (2.9)$$

With the appropriate RBF kernel $\phi(r)$, matrix $\boldsymbol{\Phi}_s$ is symmetric and positive-definite. Provided that, for relative small values of N , the matrix can be decomposed using Cholesky decomposition, which makes efficient the numerical solution of the systems.

From the equation 2.8 it is easily understood that RBF vector field is a point function with the big advantage to be independent from the mesh itself (i.e., connectivity of the CFD mesh is not required). Furthermore, the displacement field is uncoupled for each spatial direction, so each one of them can be interpolated separately, and the equation 2.8 becomes

$$\delta_{s_x}(\mathbf{x}_s) = \sum_{n=1}^N \mathbf{w}_n^{\delta_x} \phi(\|\mathbf{x}_s - \mathbf{x}_{s_n}\|) \quad (2.10)$$

$$\delta_{s_y}(\mathbf{x}_s) = \sum_{n=1}^N \mathbf{w}_n^{\delta_y} \phi(\|\mathbf{x}_s - \mathbf{x}_{s_n}\|) \quad (2.11)$$

$$\delta_{s_z}(\mathbf{x}_s) = \sum_{n=1}^N \mathbf{w}_n^{\delta_z} \phi(\|\mathbf{x}_s - \mathbf{x}_{s_n}\|) \quad (2.12)$$

And the Equation 2.9 is decomposed in three linear systems

$$\begin{aligned} \Phi W_x &= \Delta_{s_x} \\ \Phi W_y &= \Delta_{s_y} \\ \Phi W_z &= \Delta_{s_z} \end{aligned} \quad (2.13)$$

As mentioned above, these three systems can be solved using Cholesky decomposition once as Φ_s matrix remains the same. However, for large sets of source points, this approach becomes infeasible due to computational cost. A potential solution may be the use of an iterative method to solve the matrix like the conjugate gradient method [32] together with a preconditioner (e.g., incomplete Cholesky [33]) which is applied to the original matrix.

Reducing the size of the set of source nodes [34] is another possible workaround. A sub-sampling algorithm is applied to decide which source nodes will be kept, and hereupon that reduced set is used to complete the training phase and compute the network's weights. A similar approach is clustering the group of source points using the k nearest neighbors algorithm. Each cluster represents a source node. The node coordinates are computed as the average coordinates of the cluster nodes, and the node's displacement is computed as the average displacement of the cluster nodes. Compared to the sub-sampling procedure, clustering increases the computational cost. However, it may lead to higher quality deformed meshes as the whole original set of source nodes contribute to the produced system.

With the calculation of the weights from Equations 2.13 the training phase comes to an end. Next the network is utilized to compute the displacement of each internal node.

$$\delta_x(\mathbf{x}) = \sum_{n=1}^N \mathbf{w}_n^{\delta_x} \phi(\|\mathbf{x} - \mathbf{x}_{s_n}\|) \quad (2.14)$$

$$\delta_y(\mathbf{x}) = \sum_{n=1}^N \mathbf{w}_n^{\delta_y} \phi(\|\mathbf{x} - \mathbf{x}_{s_n}\|) \quad (2.15)$$

$$\delta_z(\mathbf{x}) = \sum_{n=1}^N \mathbf{w}_n^{\delta_z} \phi(\|\mathbf{x} - \mathbf{x}_{s_n}\|) \quad (2.16)$$

If M is the number of internal nodes, the displacement of which is to be computed, above equations are written in matrix form

$$\underbrace{\begin{bmatrix} \delta_1^\top \\ \vdots \\ \delta_M^\top \end{bmatrix}}_{\Delta} = \underbrace{\begin{bmatrix} \phi_1(\mathbf{x}_1) & \dots & \phi_N(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_M) & \dots & \phi_N(\mathbf{x}_M) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} \mathbf{w}_1^\top \\ \vdots \\ \mathbf{w}_N^\top \end{bmatrix}}_W \quad (2.17)$$

where Φ is a $M \times N$ matrix, the so-called interpolation matrix.

Chapter 3

Quaternions

3.1 Introduction

The concept of quaternions was realized by the Irish mathematician W.R. Hamilton in 1843. Legend says that Hamilton was on his way to the Royal Irish Academy with his wife when he suddenly came up with the idea of adding a fourth dimension in order to solve the problem of triples multiplication. Excited with his finding, as the couple passed the Broome Bridge, he carved the following equation into the stone of the bridge

$$i^2 = j^2 = k^2 = ijk = -1 \quad (3.1)$$

Even today, there is a plaque on the Broome Bridge commemorating his breakthrough of the above fundamental equation of quaternion algebra, which became the first non-commutative algebra to be studied.

There are two ways widely used to represent a rotation in three-dimensional space about an axis through the origin. A 3×3 orthogonal matrix, the so-called rotation matrix, and three angles with respect to a fixed coordinate system, the so-called Euler angles. However, the above representations have some specific drawbacks. A 3×3 rotation matrix representation seems to store redundant information as only three of its nine elements are independent. Additionally, the extraction of the rotation axis and angle requires several calculation steps, which makes the geometric interpretation of the rotation a bit fuzzy. Euler angles appear to be more memory efficient than rotation matrix since they only store three numbers and are pretty intuitive as any rotation is decomposed into three basic rotations. Nonetheless, when Euler angles are applied, three matrix multiplication is needed. Besides that,

they also suffer from the gimbal lock phenomenon, which occurs when two of the three rotational axes become aligned, leading to the loss of one rotational degree of freedom.

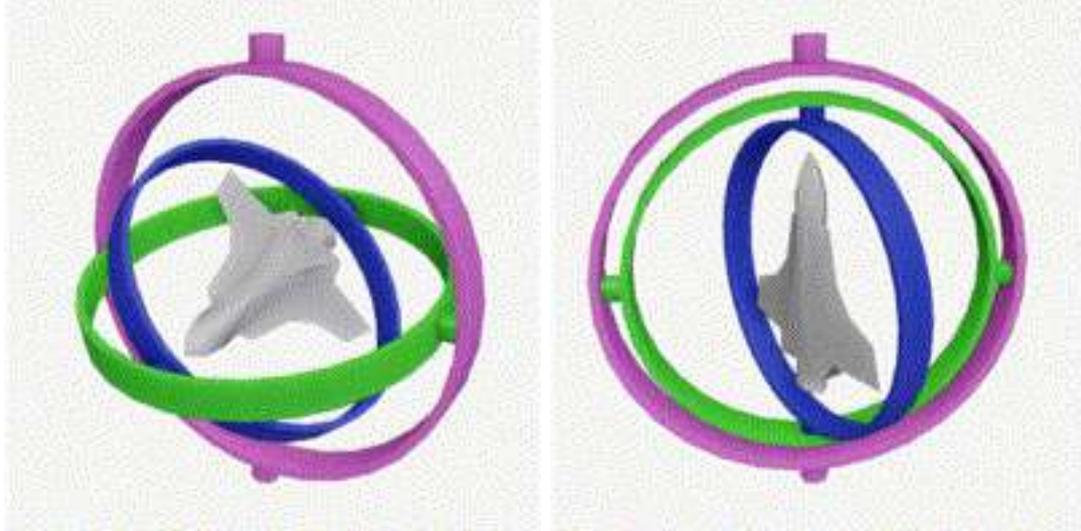


Figure 3.1: *Illustration of gimbal lock phenomenon. On the left image there is no gimbal lock, the three gimbals are independent (i.e., neither of the axis are aligned). On the right image two out of the three gimbals are in the same plane (i.e., two axis are aligned) and one degree of freedom is lost [35]*

Quaternions strike a nice balance of both being a lightweight, but free from the gimbal lock phenomenon, way to encode any rotation in three-dimensional space. A quaternion is essentially a 4-tuple, which is a much more concise representation than a rotation matrix, and the rotation is applied directly contrary to Euler angles. These make quaternions ideal for representing rotations. Furthermore, their geometric interpretation is quite distinct as the rotation axis, and angle can be almost directly recovered.

3.2 Quaternions Background

Quaternions can be regarded as 3D extension of complex numbers. Mathematically, a quaternion, is a four-element vector which is composed of one real and three imaginary numbers. The set of quaternions, with the addition and multiplication operators, form a non-commutative ring which is denoted as \mathbb{H} . The standard orthonormal basis for \mathbb{R}^3 is given by three imaginary unit vectors

$$\mathbf{i} = (1, 0, 0), \quad \mathbf{j} = (0, 1, 0), \quad \mathbf{k} = (0, 0, 1)$$

which also satisfy the following rules introduced by Hamilton

$$\begin{aligned}
ii &= jj = kk = ijk = -1 \\
ij &= -ji = k \\
jk &= -kj = i \\
ki &= -ik = j
\end{aligned} \tag{3.2}$$

A quaternion Q can then be expressed with the following form

$$Q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}, \quad q_\alpha \in \mathbb{R}, Q \in \mathbb{R}^4 \tag{3.3}$$

Similarly with complex numbers, a quaternion, Q , can be divided into a real and an imaginary part. Real part is the scalar part q_0 of the quaternion and the purely imaginary part is the vector part $\mathbf{q} = (q_1, q_2, q_3)$ which is expressed like a vector in the three-dimensional space.

$$Q = [q_0, \mathbf{q}] \tag{3.4}$$

3.2.1 Quaternion Arithmetic

Quaternions addition is component wise similar to complex numbers. Consider the quaternions Q, P , then

$$Q + P = (q_0 + p_0) + (q_1 + p_1)\mathbf{i} + (q_2 + p_2)\mathbf{j} + (q_3 + p_3)\mathbf{k} \tag{3.5}$$

It is pretty obvious that subtraction is derived by adding to Q the negative quaternion of P ($-P$) with components $-p_\alpha$, $\alpha = 0, 1, 2, 3$

$$Q - P = (q_0 - p_0) + (q_1 - p_1)\mathbf{i} + (q_2 - p_2)\mathbf{j} + (q_3 - p_3)\mathbf{k} \tag{3.6}$$

Product of two quaternions, called Hamilton product, is determined by the product of the basis elements 3.2 and the distributive law. The product is given as

$$\begin{aligned}
QP &= q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3 \\
&+ (q_0p_1 + q_1p_0 + q_2p_3 - q_3p_2)\mathbf{i} \\
&+ (q_0p_2 - q_1p_3 + q_2p_0 + q_3p_1)\mathbf{j} \\
&+ (q_0p_3 + q_1p_1 - q_2p_1 + q_3p_0)\mathbf{k}
\end{aligned} \tag{3.7}$$

Exploiting the vector dot and vector cross product in \mathbb{R}^3 , together with Equation 3.4, Equation 3.7 can be written in a more concise form

$$\begin{aligned} Q \times P &= [q_0, \mathbf{q}] [p_0, \mathbf{p}] \\ &= q_0 p_0 - \mathbf{q} \cdot \mathbf{p} + q_0 \mathbf{p} + p_0 \mathbf{q} + \mathbf{q} \times \mathbf{p} \end{aligned} \quad (3.8)$$

where \cdot denotes the vector dot product, and \times denotes the vector cross product.

From Equation 3.8 is easily derived that quaternion multiplication is not generally commutative. Therefore, instead of a divide operation a multiplication of Q by the inverse P^{-1} is used.

$$Q \div P = QP^{-1} \quad (3.9)$$

where P^{-1} is given

It is clear that the product of two quaternions is also a quaternion. The same is applicable to addition and subtraction. Therefore, the set \mathbb{H} of quaternions is closed under addition, subtraction and multiplication.

Based on the above, quaternions have some very useful properties which are presented bellow and will be used later

- Associative: $(Q_1 Q_2) Q_3 = Q_1 (Q_2 Q_3)$
- Distributive: $Q_1 (Q_2 + Q_3) = Q_1 Q_2 + Q_1 Q_3$
- Not commutative: $Q_1 Q_2 \neq Q_2 Q_1$
- Conjugate of a quaternion $Q^* = q_0 - q_1 i - q_2 j - q_3 k$
- Magnitude of a quaternion $\|Q\| = \sqrt{Q Q^*} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$
- Inverse of a quaternion: $Q^{-1} = Q^* / (Q Q^*)$

A quaternion Q with real part equal to zero ($q_0 = 0$) is called a pure quaternion $Q = [0, \mathbf{v}]$. Thus a standard 3D vector (i.e., a point in \mathbb{R}^3) can be readily stored in a pure quaternion. On the other hand, a real quaternion is a quaternion with a zero vector term ($\mathbf{q} = 0$), and the product of two such quaternions is also a real quaternion.

3.3 Quaternions and spatial rotation

As already mentioned, quaternions are ideal for modeling rotations. In three dimensional space, according to Euler's rotation theorem, any rotation or sequence of rotations of a rigid body or coordinate system about a fixed point is equivalent to a single rotation by a given angle θ about a fixed axis, the so-called *Euler axis*, that runs through the fixed point. The Euler axis is typically represented by a unit vector \mathbf{u} . Therefore, any rotation in three dimensions can be regarded as a combination of a scalar θ and a vector \mathbf{u} .

Quaternions are a simple way to encode this axis-angle entity. They can be used to apply the corresponding rotation to a position vector (a point relative to the origin in \mathbb{R}^3). The vector part of a quaternion represents the axis around which rotation occurs, and the scalar part represents the magnitude of the rotation.

A rotation of a point, $p = (x, y, z)$, about a unit vector, u , by a certain angle, θ , using quaternions consists of three steps. At first, a pure quaternion P is constructed expressing the point p as

$$P = [0, \mathbf{p}] \tag{3.10}$$

A quaternion Q representing the angle and the axis of rotation is composed as

$$\begin{aligned} Q &= [s, \mathbf{v}], \quad s = \cos \frac{\theta}{2}, \quad \mathbf{v} = \mathbf{u} \sin \frac{\theta}{2} \quad \text{or} \\ Q &= \left[\cos \frac{\theta}{2}, \mathbf{u} \sin \frac{\theta}{2} \right] \end{aligned} \tag{3.11}$$

Figure 3.2 depicts the physical interpretation of such a quaternion that its components represent the axis of rotation \mathbf{u} and rotation angle θ . The quaternion essentially represents the orientation of Frame B with respect to Frame A.

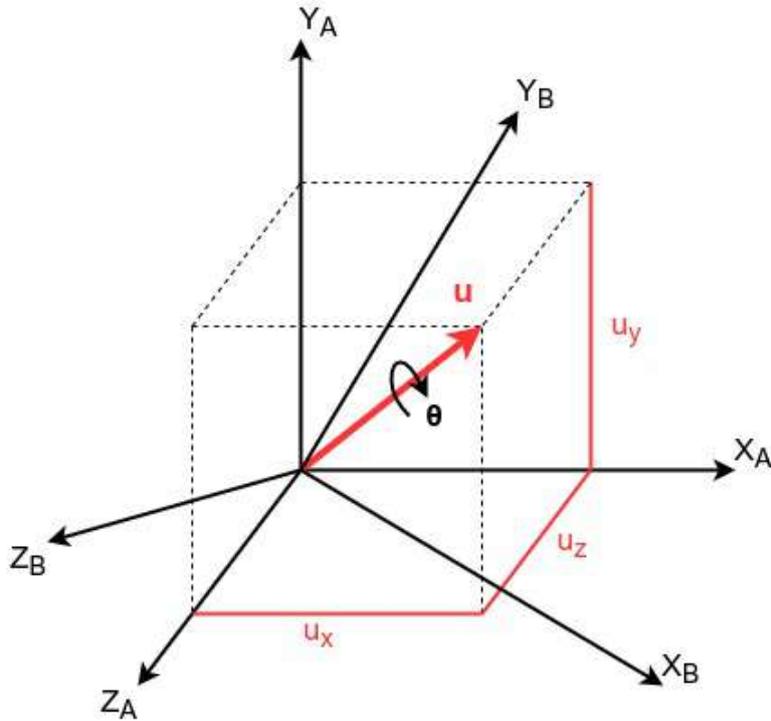


Figure 3.2: A quaternion representing a rotation from Frame A to Frame B. The components of this quaternion are related to the rotation axis \mathbf{u} and angle θ according to Equation 3.11.

Finally, the following equation produces the pure quaternion expressing the rotated point $P_{rotated} = [0, \mathbf{P}_{rotated}]$

$$P_{rotated} = QPQ^{-1} \quad (3.12)$$

If the axis of rotation \mathbf{u} is normalized ($\|\mathbf{u}\| = 1$) then $\|Q\| = 1$ and Q is a unit quaternion. Exploiting that, Equation 3.12 can be written in a much more convenient form. From the above set of quaternions properties, it is clear that the conjugate of a quaternion is obtained more cost-efficiently than its inverse.

$$P_{rotated} = QPQ^* \quad (3.13)$$

From the above set of quaternions properties, it is clear that the conjugate of a quaternion is obtained much more cost-efficiently than its inverse. Actually, it is a commonly used technique to normalize every quaternion that is used to rotate a vector. The formulas for using quaternions to rotate a vector implicitly assume the quaternion is of unit magnitude. The errors that result from such operations using a non-normalized quaternion are proportional to the square of quaternion's magnitude; hence quadratic error growth is to be avoided. Besides, from a normalized

quaternion, the angle and the axis of rotation are extracted directly from its elements as it is demonstrated in Equation 3.11. It should be noted that any operation that theoretically produces a unit quaternion will still need to be followed from a normalization as floating-point precision errors will cause it to not be unit length. A quaternion can be normalized just like a four dimensional vector by dividing each of the four components by the Euclidean norm.

$$Q_{normalized} = \frac{q_0}{\|Q\|} + \frac{q_1}{\|Q\|} + \frac{q_2}{\|Q\|} + \frac{q_3}{\|Q\|} \quad (3.14)$$

Quaternions also simplify the composition of multiple rotations. Taking into consideration that quaternions are associative but not commutative and the definition of the inverse of a quaternion, the composite rotation information is expressed by the following multiplication

$$Q_{comp} = Q_n Q_{n-1} \dots Q_2 Q_1 \quad (3.15)$$

while the subscript also indicates the order of rotation. For example, if Q_1 and Q_2 are unit quaternions representing two successive rotations of point \mathbf{p} ($P = [0, \mathbf{p}]$), the two rotations can be combined as

$$\begin{aligned} P_{rotated} &= Q_2 (Q_1 P Q_1^*) Q_2^* \\ &= (Q_2 Q_1) P (Q_1^* Q_2^*) \\ &= (Q_2 Q_1) P (Q_1 Q_2)^* \end{aligned} \quad (3.16)$$

Quaternion coordinates represent rotation as Cartesian coordinates represent translation as a single vector.

Chapter 4

The proposed hybrid approach

In this chapter the proposed hybrid method is presented. The goal is to generate deformed meshes using undeformed meshes and configurations related to the deformation. There are mainly two parts in the hybrid procedure that is proposed in this diploma thesis. The first part consists in the RBF mesh morphing as presented in Section 2.3. In the second part quaternions are determined using comparison of cell orientation and displacement on boundaries, whose deformed configurations are known. Then, a weighting function is utilized in order to couple the above parts and propagate the displacements to volume.

4.1 Construction of quaternions on boundaries

The quaternions which would be used together with the RBF mesh morphing method are defined uniquely on boundary nodes. Samareh [4] defined a quaternion by calculating orientation on each cell of undeformed and deformed airfoils. He expressed the displacement of an arbitrary node on a body as a sum of a rotation and translation and then these two components of transformation information are propagated to the space nodes to define their displacements values. For the construction of the boundary quaternions the same approach was used. If \mathbf{x}_u , \mathbf{x}_d are the position vectors of an undeformed node and its corresponding deformed node respectively and \mathbf{t} a translation vector, then the division of a node's displacement into a rotation and a translation is shown bellow

$$\mathbf{x}_d = \text{rotation of } \mathbf{x}_u + \mathbf{t} \quad (4.1)$$

Using quaternion algebra, Equation 4.1 can be written as follows

$$X_d = QX_uQ^* + T \quad (4.2)$$

where $X_u = [0, \mathbf{x}_u]$ and $X_d = [0, \mathbf{x}_d]$ are pure quaternions representing position vectors. Q is the rotation quaternion and $T = [0, \mathbf{t}]$ is a pure quaternion expressing the translation vector. The deformation information is stored in the rotation and the translation quaternions. They are determined by geometric considerations which emerge from the comparison of the segments (or cells in 3D) orientation and the displacement between undeformed and deformed shapes of boundaries.

The process of determining boundary quaternions is conducted for each node on the boundaries, while on fixed boundaries, Q, T are defined as $Q = [1, \mathbf{0}], T = [0, \mathbf{0}]$. It can be divided into three steps as described in [4]. In the first step, the specific mesh point of the undeformed boundary x_{uc} and its corresponding point of the deformed boundary x_{dc} are translated to the origin. Following that, normal vectors ($\mathbf{n}_u, \mathbf{n}_d$) are defined on the node using segment or cell composed by neighboring nodes.

In the second step, the undeformed segment is rotated so that the undeformed boundary normal vector n_u aligns with the deformed boundary normal vector n_d . This rotation is modeled with a quaternion Q_1 . The axis of rotation is determined by the normal vector of a plane shared by both undeformed and deformed normal vectors and is given by the cross product of those two. As noted before, it is crucial every operation that produce a vector or a quaternion to be followed by a normalization. Thus, the normal vector \mathbf{u}_1 of the shared plane is normalized before quaternion construction which leads the quaternion Q_1 to be unit length.

$$\mathbf{u}_1 = \mathbf{n}_u \times \mathbf{n}_d \quad (4.3)$$

The angle of rotation is the angle θ_1 between the two normal vectors ($\mathbf{n}_u, \mathbf{n}_d$).

$$\theta_1 = \cos^{-1}(\mathbf{n}_u, \mathbf{n}_d) \quad (4.4)$$

Having determined both axis and angle a quaternion Q_1 is constructed as

$$Q_1 = \left[\cos \frac{\theta_1}{2}, \mathbf{u}_1 \sin \frac{\theta_1}{2} \right] \quad (4.5)$$

The visual interpretation of the above process for a two-dimensional mesh is presented in Figure 4.1.

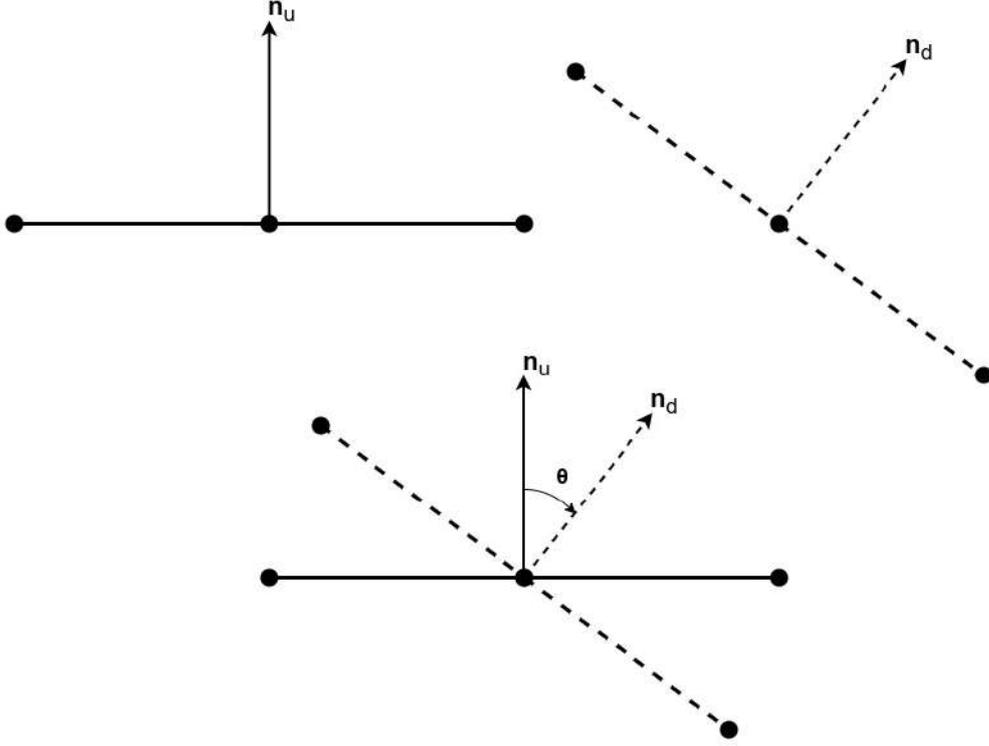


Figure 4.1: *Process of boundary quaternion construction in a segment of a two-dimensional mesh.*

In three-dimensional cases, a third step is required. The angle between vectors from the cell center (origin) to the cell vertices (neighbors of the node) of the undeformed and deformed cells should be minimized. This is achieved through a rotation around the deformed boundary normal vector n_d about an optimal angle θ_2 . This angle is defined as the average angle between corresponding edges of the rotated undeformed boundary (the boundary that arose from previous steps) and the edges of the deformed boundary.

$$\theta_2 = \frac{\sum_{i=1}^{N_{vertex}} [(\mathbf{x}_{u,i} - \mathbf{x}_{uc}) (\mathbf{x}_{d,i} - \mathbf{x}_{dc})]}{N_{vertex}} \quad (4.6)$$

The second rotation is also modeled with a quaternion Q_2 , which is constructed using \mathbf{n}_d as axis and angle θ_2 .

$$Q_2 = \left[\cos \frac{\theta_2}{2}, \mathbf{n}_d \sin \frac{\theta_2}{2} \right] \quad (4.7)$$

The two rotations can be represented with one quaternion using Equation 3.15.

$$Q = Q_2 Q_1 \quad (4.8)$$

The procedure to obtain the two rotations which finally form the composed quaternion Q is depicted in Figure 4.2.

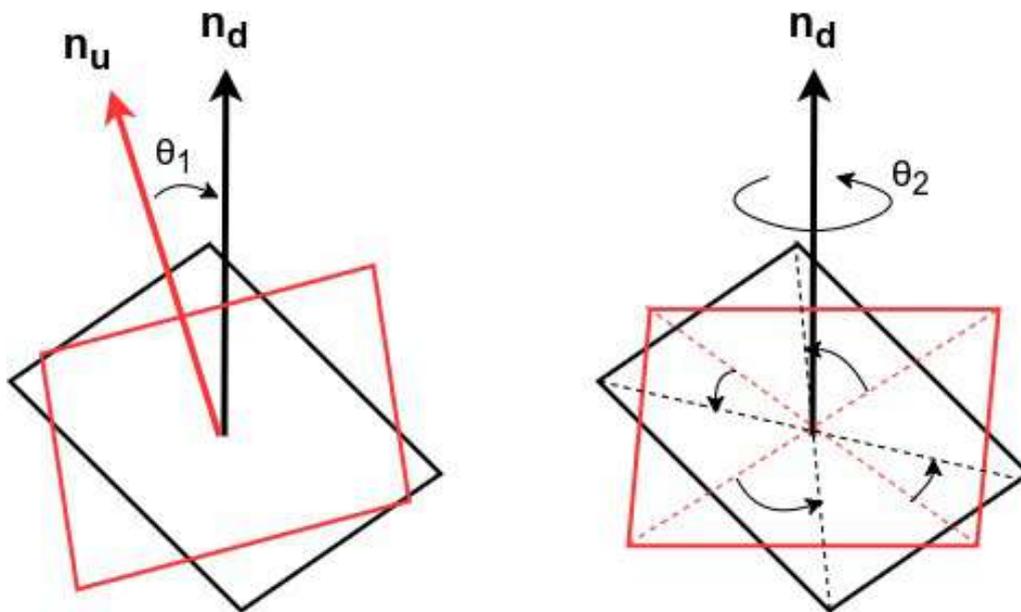


Figure 4.2: *Process of boundary quaternions construction in a cell of a three-dimensional mesh.*

After the rotation parts have been determined, the pure quaternion T representing the total translation vector for the boundary can be defined as

$$T = X_d - QX_uQ^* \quad (4.9)$$

4.2 Method description

This proposed hybrid approach couples the aforementioned RBF mesh morphing method with quaternion algebra. As described above, quaternions are defined on the boundary nodes, or on the nodes where the displacement is known, modeling any nodal movement as a sum of a rotation and a translation. The known displacements are also used to train the RBF network. Following that, every node of the domain moves according to the move of its closest boundary node (quaternions) and the move of the whole boundary (RBF). Therefore, each internal displacement is the weighted sum of the displacement of its closest boundary node and the displacement

as emerges normally from RBF method.

$$\delta_{internal} = \gamma \delta_{RBF} + (1 - \gamma) \delta_{closest\ node} \quad (4.10)$$

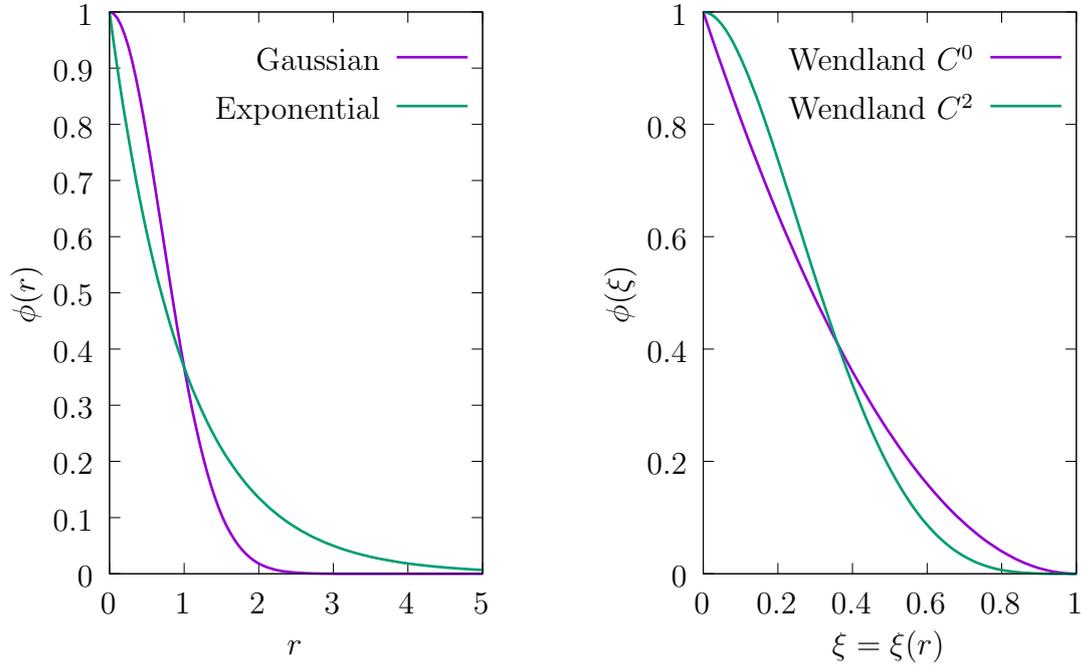
where γ is the weight of each displacement and it is a function of distance from boundary d . So

$$\gamma = \gamma(d) \quad (4.11)$$

Distance from boundary is defined as the distance from the closest boundary node. Weight function is deployed in such way that quaternions have high influence near the boundary which erodes as the distance from boundary increases. Several functions are suitable to be utilized as weight function. The main condition is the output of the function to decrease smoothly as distance increases. Such functions are presented bellow.

Name	$\gamma(d)$
Gaussian	$e^{-(\varepsilon d)^2}$
Exponential	$e^{-\varepsilon d}$
Wendland C^0	$(1 - \xi)^2$
Wendland C^2	$(1 - \xi)^4 (4\xi + 1)$

Table 4.1: *Possible weighting functions*



(a) Gaussian and Exponential functions with $\varepsilon = 1$ deployed as weighting functions

(b) Wendland C^0 and Wendland C^2 deployed as weighting functions with global support

Figure 4.3: Functions that are suitable to be utilized as weighting functions since their output decreases smoothly as the distance r increases

Shape-parameter ε is used to control the smoothness of the transition from the dominance of the method with quaternions to the dominance of the RBF method. Here, Wendland functions that were described in Section 2.2 as radial basis functions with local support, are deployed as global support functions. This is done by defining ξ as the ratio of current distance from boundary to the maximum distance from the boundary that appears in the whole mesh domain.

$$\xi = \frac{d}{d_{max}} \quad (4.12)$$

Chapter 5

Application to the mesh around a rotated sycamore leaf

5.1 Problem description

In this chapter, the RBF method and the proposed hybrid method are applied to deform the mesh around a rotated sycamore leaf located within a stationary circle. Between the leaf and the outer circle, a mesh with triangular elements is generated using the advanced-front method [36] with a software of PCOpt/NTUA. The mesh is provided from PCOpt/NTUA in two files (i.e., LTT format: one file with nodes coordinates and one file with nodes connectivity), and it is then transformed in VTK file format in order to be compatible with ParaView [37]. For this purpose, software that converts both 2D and 3D LTT unstructured meshes into VTK format was developed. Additionally, every element of the mesh is colored according to its shape quality (see Appendix A) through a color-quality mapping. Figure 5.1 depicts the original mesh as it is visualized in ParaView.

The sycamore leaf rotates around the origin, and subsequently, the internal mesh is deformed, conforming to the updated boundary nodes locations. The aim is to test both methods with various activation functions and compare the resulting deformed meshes. Although this application seems pretty simple and irrelevant with the CFD field, the complexity of the leaf's geometry can provide some quite informative results.

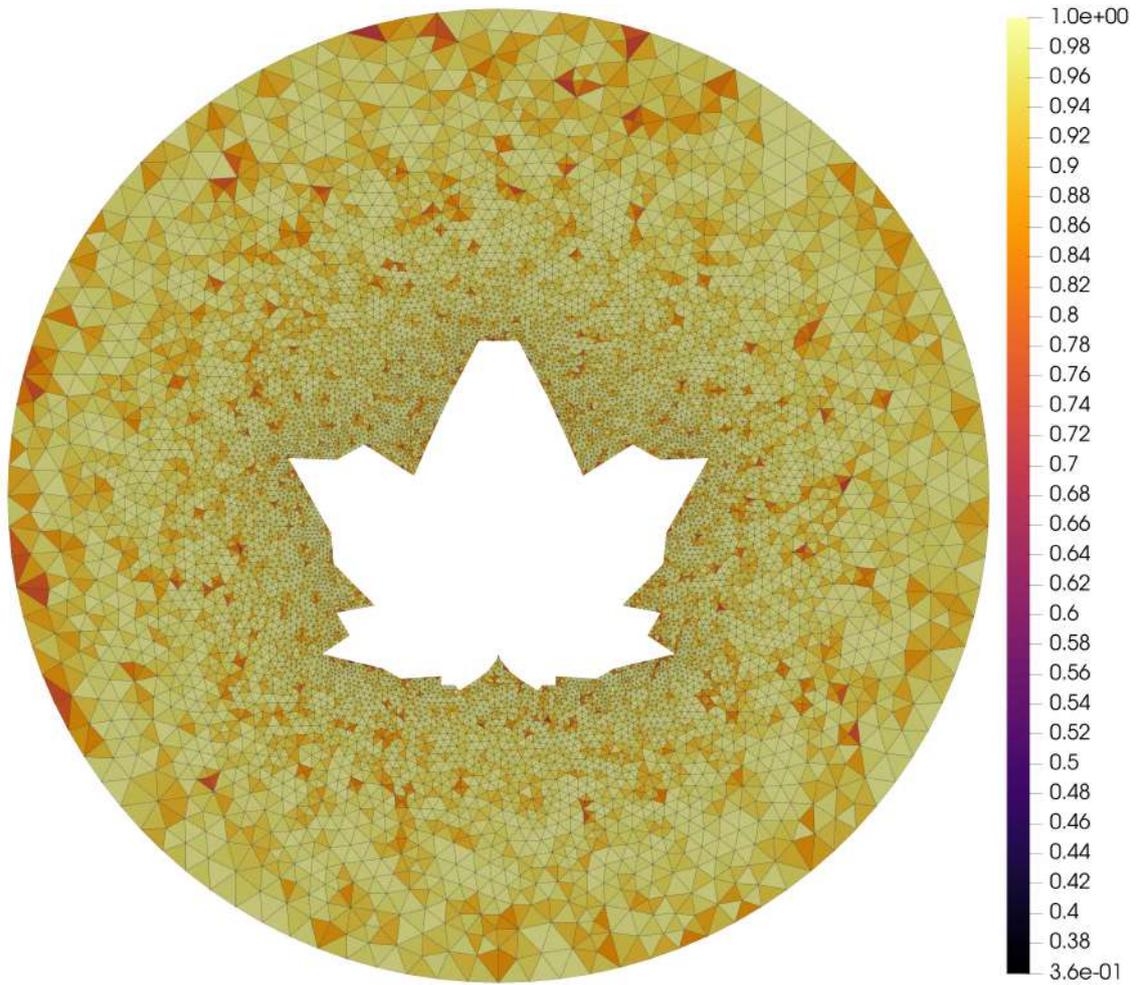


Figure 5.1: *The original mesh with triangular elements around a sycamore leaf as it was provided from software of PCOpt/NTUA*

Two radial basis functions with global support and one with compact support are chosen as activation functions of the RBF network. Globally supported inverse quadratic kernel

$$\frac{1}{1 + (\varepsilon r)^2} \quad (5.1)$$

and inverse multi-quadric kernel [38]

$$\frac{1}{\sqrt{1 + (\varepsilon r)^2}} \quad (5.2)$$

are employed, where ε is the shape-parameter regulating the decay of the kernel. As compactly supported RBF, Wendland C^0 (Equation 5.3) and Wendland C^2 (Equa-

tion 5.4) [31] were preferred

$$\phi(\xi) = \begin{cases} (1 - \xi)^2 & \text{if } \xi \leq 1 \\ 0 & \text{if } \xi > 1 \end{cases} \quad (5.3)$$

$$\phi(\xi) = \begin{cases} (1 - \xi)^4 (4\xi + 1) & \text{if } \xi \leq 1 \\ 0 & \text{if } \xi > 1 \end{cases} \quad (5.4)$$

where $\xi = \frac{r}{r_s}$. For the compact support RBF the radius of influence r_s , was selected thus every mesh node of the domain will be influenced by at least one source node. Therefore, it is equal to the maximum distance between a domain node and its closest boundary node.

5.2 Results

At first the original RBF method will be tested using two activation functions of global support and two activation functions of local support as already described previously. For the compactly supported RBFs various support radius will be tested. The RBF that achieves the best results in relation to the quality of the deformed mesh and the angle of rotation at which inverted elements appear will be used subsequently in the hybrid method. The latter will be tested with the weighting functions that were presented in Section 4.2.

5.2.1 Inverse quadratic

In this section the globally supported inverse quadratic function (Equation 5.1) was chosen as activation function of the RBF network. The shape-parameter, ε , was selected to be equal to 1 in order to achieve a smooth decay of the kernel. The sycamore leaf is repeatedly rotated in increasing angles until inverted elements appear. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each angle of rotation are presented in Table 5.1.

$\phi(\mathbf{r})$	5	10	15	20
$\overline{\mathbf{f}}_{\text{ss}}$	0.934	0.897	0.850	0.798
$\sigma_{\mathbf{f}_{\text{ss}}}$	0.052	0.083	0.127	0.176
$\mathbf{f}_{\text{ss},\text{min}}$	0.359	0.359	0.220	0.043

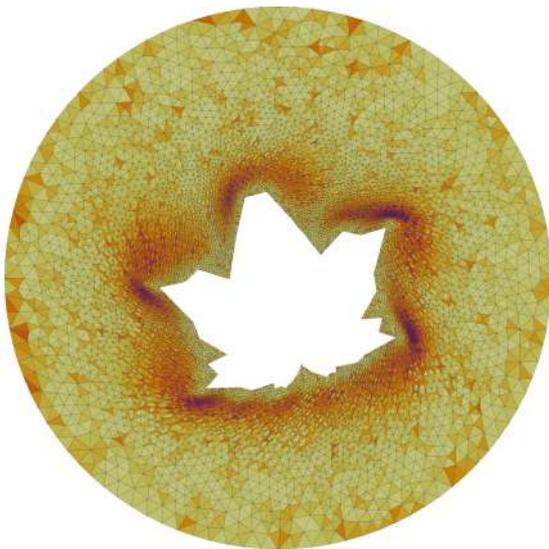
Table 5.1: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of sycamore leaf. Inverse quadratic is the selected RBF kernel



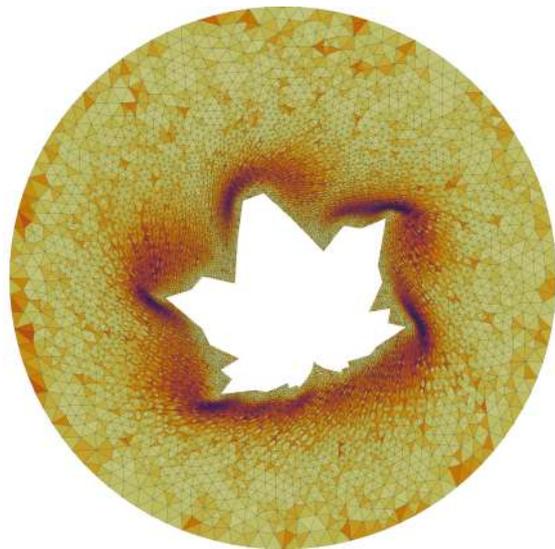
(a) 5 degrees angle of rotation



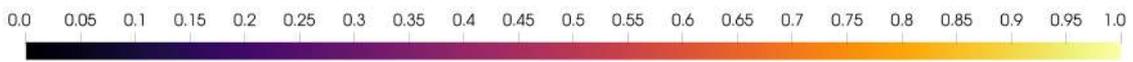
(b) 10 degrees angle of rotation



(c) 15 degrees angle of rotation



(d) 20 degrees angle of rotation



(e) Size-shape metric

Figure 5.2: Application of the typical RBF mesh deformation method around a sycamore leaf. The leaf rotates 5, 10, 15 and 20 degrees around the origin. Inverse quadratic function was chosen as the activation function of the RBF network. Colour corresponds to size-shape metric f_{ss} . For angles of rotation greater than 22 degrees inverted elements appear

5.2.2 Inverse multi-quadric

Next, the globally supported inverse multi-quadric function (Equation 5.2) was chosen as activation function of the RBF network. The shape-parameter, ε , was selected to be equal to 1. Similar to the previous section, the sycamore leaf is repeatedly rotated in increasing angles until inverted elements appear. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric, which is introduced in Appendix A, for each angle of rotation are presented in Table 5.2.

ϕ (degrees)	15	20	25	30
$\overline{f_{ss}}$	0.887	0.849	0.807	0.762
$\sigma_{f_{ss}}$	0.087	0.118	0.152	0.187
$f_{ss,\min}$	0.359	0.295	0.196	0.080

Table 5.2: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of sycamore leaf. Inverse multi-quadric is the selected RBF kernel.

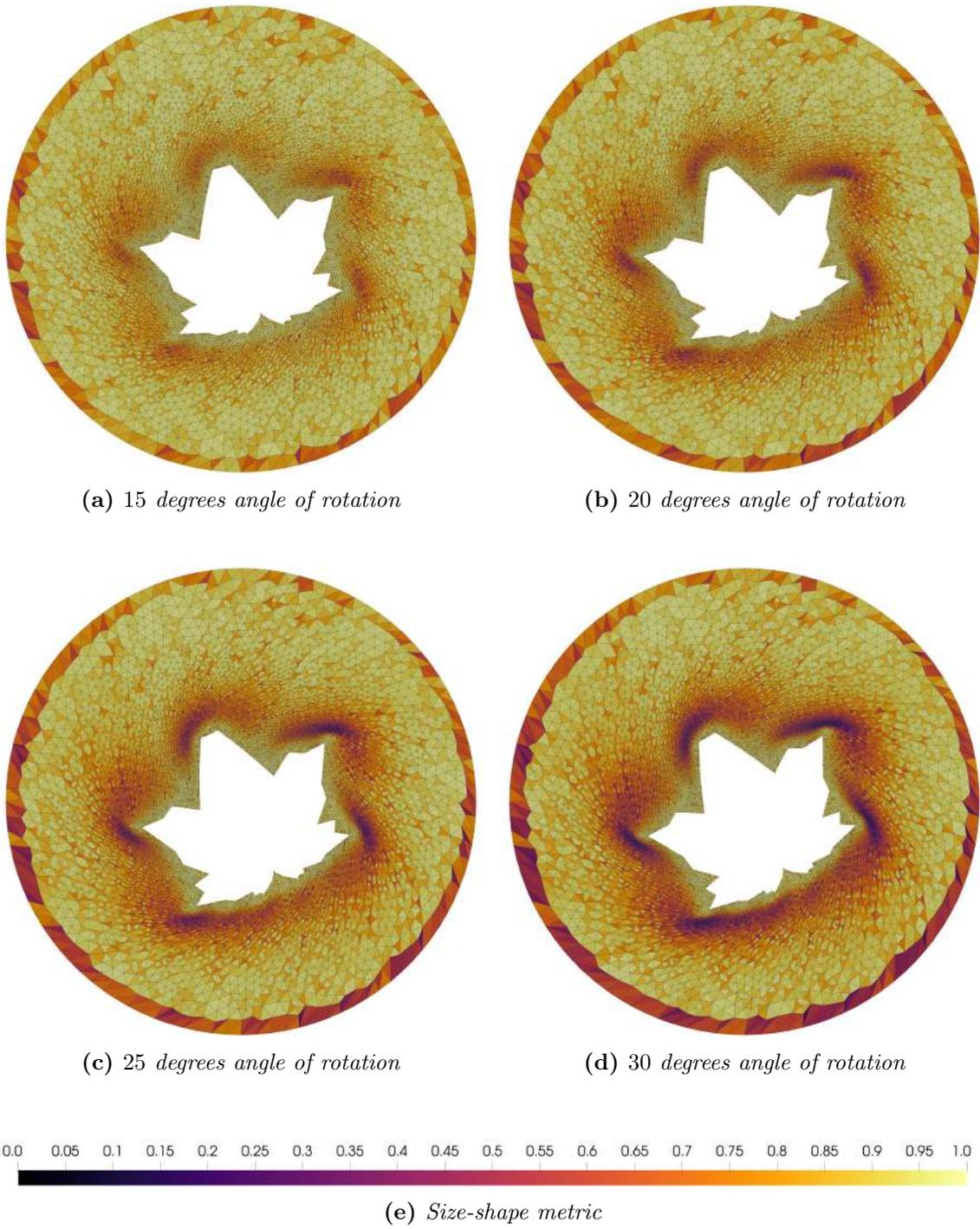


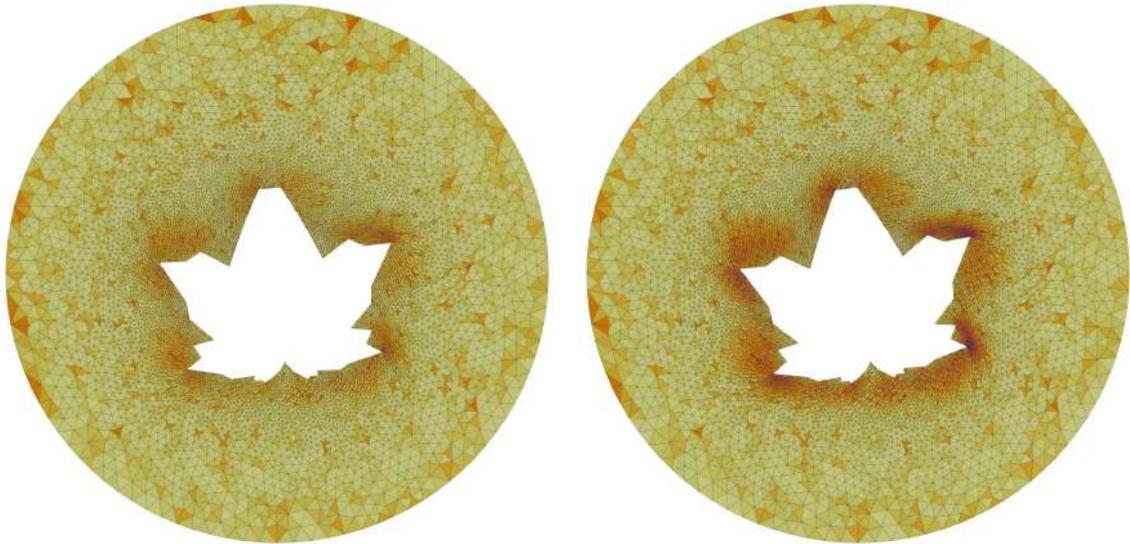
Figure 5.3: Application of the typical RBF mesh deformation method around a sycamore leaf. The leaf rotates 15, 20, 25 and 30 degrees around the origin. Inverse multi-quadric function was chosen as the activation function of the RBF network. Colour corresponds to size-shape metric f_{ss} . For angles of rotation greater than 33 degrees inverted elements appear.

5.2.3 Wendland C^0 ($R_s = 2$)

Following the above functions with global support, in this section compactly supported function Wendland C^0 is selected as RBF kernel. At first a relative small support radius, $R_s = 2$ is used. The procedure is the same as previously. In Table 5.3 the average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric for three angles of rotations are presented. As the support radius R_s is relative small, for angles greater than 12 degrees the deformed mesh develops inverted elements.

ϕ (degrees)	5	8	10
$\overline{f_{ss}}$	0.918	0.881	0.854
$\sigma_{f_{ss}}$	0.070	0.108	0.138
$f_{ss,\min}$	0.359	0.239	0.067

Table 5.3: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of sycamore leaf. Wendland C^0 with support radius $R_s = 2$ is the selected RBF kernel.

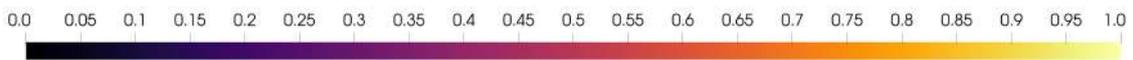


(a) 5 degrees angle of rotation

(b) 8 degrees angle of rotation



(c) 10 degrees angle of rotation



(d) Size-shape metric

Figure 5.4: Application of the typical RBF mesh deformation method around a sycamore leaf. The leaf rotates 5, 8 and 10 degrees around the origin. Wendland C^0 function with support radius $R_s = 2$ was chosen as the activation function of the RBF network. Colour corresponds to size-shape metric f_{ss} . For angles of rotation greater than 12 degrees inverted elements appear.

5.2.4 Wendland C^0 ($R_s = 5$)

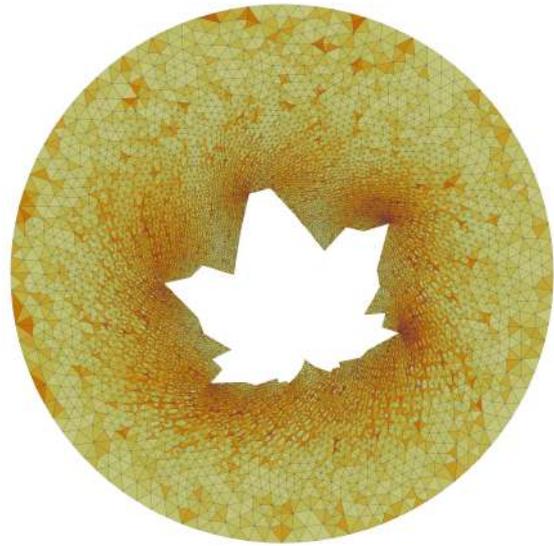
In this section compactly supported function Wendland C^0 is selected once more as RBF kernel but with a greater support radius $R_s = 5$. The procedure is the same as previously. In Table 5.4 the average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric for three angles of rotations are presented. As the support radius R_s is relative small, for angles greater than 26 degrees the deformed mesh develops inverted elements.

ϕ (degrees)	10	15	20	25
$\overline{f_{ss}}$	0.908	0.867	0.819	0.765
$\sigma_{f_{ss}}$	0.065	0.091	0.122	0.157
$f_{ss,\min}$	0.359	0.359	0.253	0.046

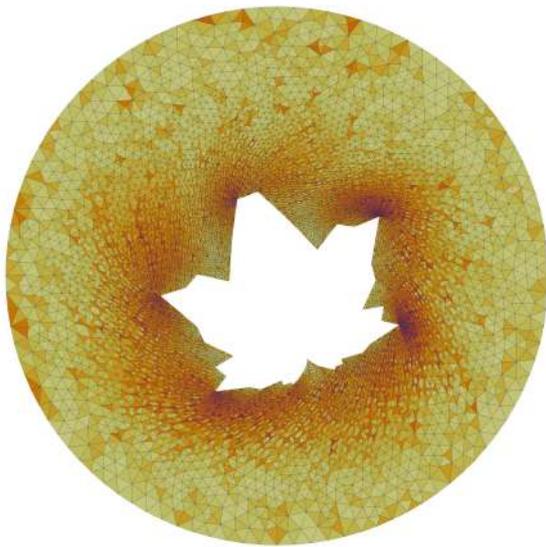
Table 5.4: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of sycamore leaf. Wendland C^0 with support radius $R_s = 5$ is the selected RBF kernel.



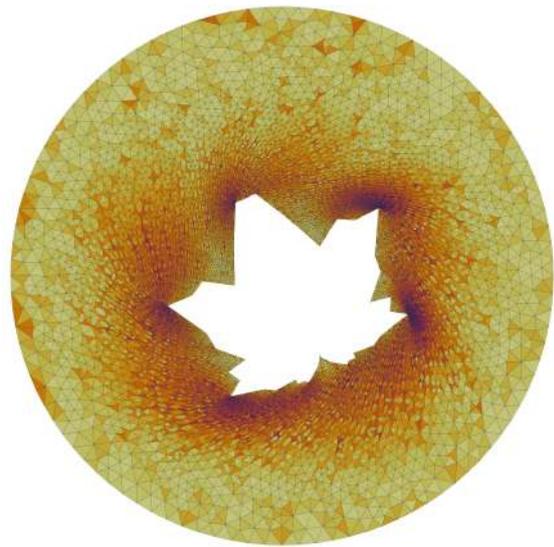
(a) 10 degrees angle of rotation



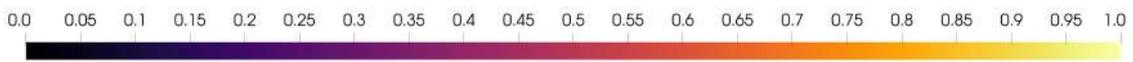
(b) 15 degrees angle of rotation



(c) 20 degrees angle of rotation



(d) 25 degrees angle of rotation



(e) Size-shape metric

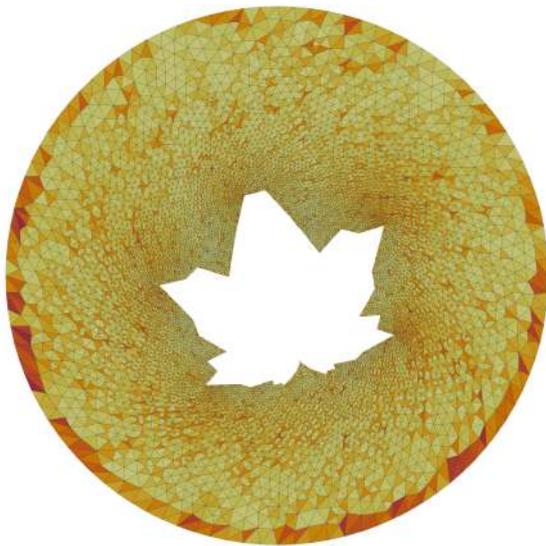
Figure 5.5: Application of the typical RBF mesh deformation method around a sycamore leaf. The leaf rotates 10, 15, 20 and 25 degrees around the origin. Wendland C^0 function with support radius $R_s = 5$ was chosen as the activation function of the RBF network. Colour corresponds to size-shape metric f_{ss} . For angles of rotation greater than 26 degrees inverted elements appear.

5.2.5 Wendland C^0 ($R_s = 11$)

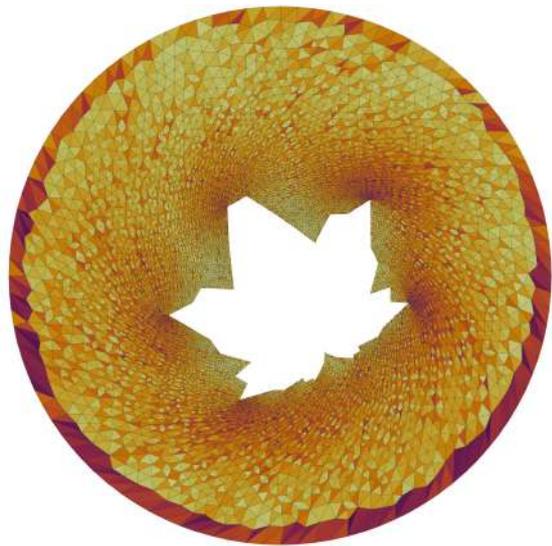
Lastly, Wendland C^0 is selected as RBF kernel but with a relative big support radius $R_s = 11$. The procedure is the same as previously. In Table 5.5 the average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric for three angles of rotations are presented. For angles greater than 48 degrees the deformed mesh presents inverted elements.

ϕ (degrees)	15	30	40	45
$\overline{f_{ss}}$	0.911	0.811	0.724	0.677
$\sigma_{f_{ss}}$	0.064	0.113	0.153	0.175
$f_{ss,\min}$	0.359	0.276	0.133	0.034

Table 5.5: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of sycamore leaf. Wendland C^0 with support radius $R_s = 11$ is the selected RBF kernel.



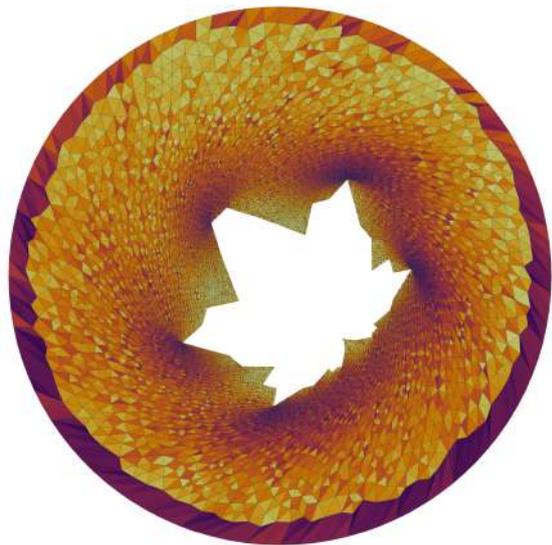
(a) 15 degrees angle of rotation



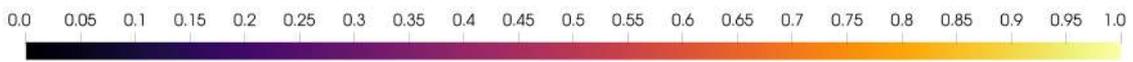
(b) 30 degrees angle of rotation



(c) 40 degrees angle of rotation



(d) 45 degrees angle of rotation



(e) Size-shape metric

Figure 5.6: Application of the typical RBF mesh deformation method around a sycamore leaf. The leaf rotates 15, 30, 40 and 45 degrees around the origin. Wendland C^0 function with support radius $R_s = 11$ was chosen as the activation function of the RBF network. Colour corresponds to size-shape metric f_{ss} . For angles of rotation greater than 48 degrees inverted elements appear.

5.2.6 Wendland C^2 ($R_s = 11$)

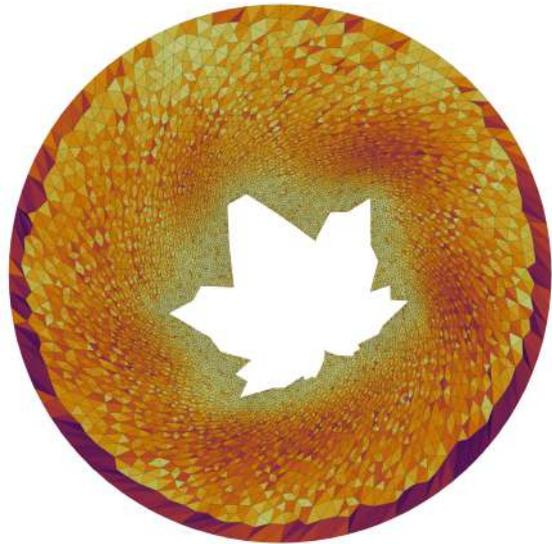
Following Wendland C^0 , in this section the also compactly supported Wendland C^2 is utilized as RBF kernel. As is shown above, the bigger the radius of support the better the quality of the deformed mesh. Therefore, for this case a relatively big radius of support, $R_s = 11$ is applied from the beginning. Similarly to previous, in Table 5.6 the average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric for three angles of rotations are presented. For angles greater than 69 degrees the deformed mesh appears inverted elements.

ϕ (degrees)	15	30	45	60
$\overline{f_{ss}}$	0.922	0.843	0.734	0.624
$\sigma_{f_{ss}}$	0.064	0.120	0.191	0.260
$f_{ss,\min}$	0.359	0.177	0.088	0.054

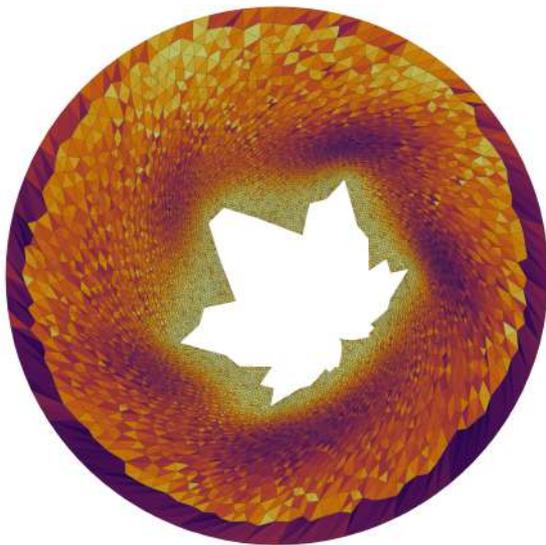
Table 5.6: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of sycamore leaf. Wendland C^2 with support radius $R_s = 11$ is the selected RBF kernel.



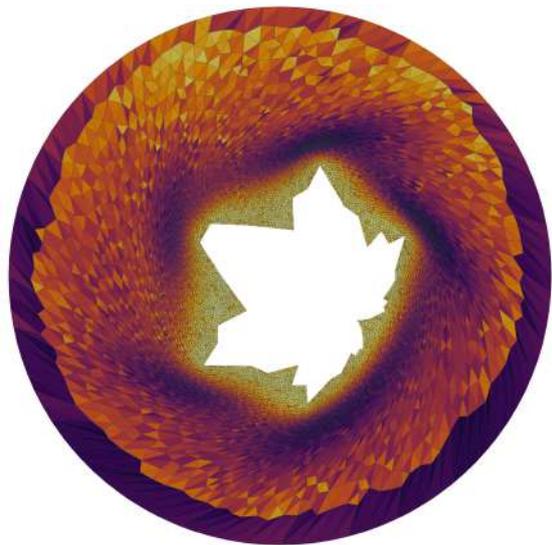
(a) 15 degrees angle of rotation



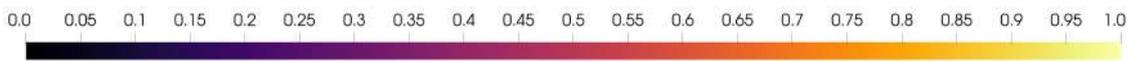
(b) 30 degrees angle of rotation



(c) 45 degrees angle of rotation



(d) 60 degrees angle of rotation



(e) Size-shape metric

Figure 5.7: Application of the typical RBF mesh deformation method around a sycamore leaf. The leaf rotates 15, 30, 45 and 60 degrees around the origin. Wendland C^2 function with support radius $R_s = 11$ was chosen as the activation function of the RBF network. Colour corresponds to size-shape metric f_{ss} . For angles of rotation greater than 69 degrees inverted elements appear.

5.2.7 Wendland C^2 ($R_s = 11$) with Gaussian ($\varepsilon = 0.5$) as weighting function

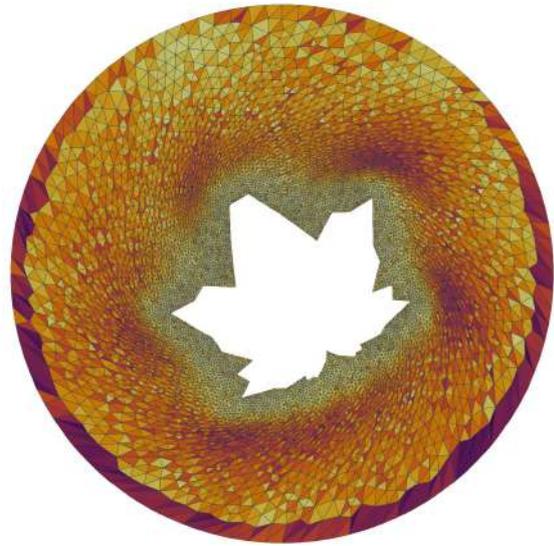
At this stage the proposed hybrid method is applied. Wendland C^2 function with a support radius $R_s = 11$ is used as the activation function of the RBF network. Gaussian function with a shape-parameter $\varepsilon = 0.5$ is used as the weighting function. In Table 5.7 the average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric for three angles of rotations are presented.

ϕ (degrees)	15	30	45	55
$\overline{f_{ss}}$	0.919	0.839	0.738	0.671
$\sigma_{f_{ss}}$	0.069	0.139	0.225	0.279
$f_{ss,\min}$	0.359	0.177	0.088	0.010

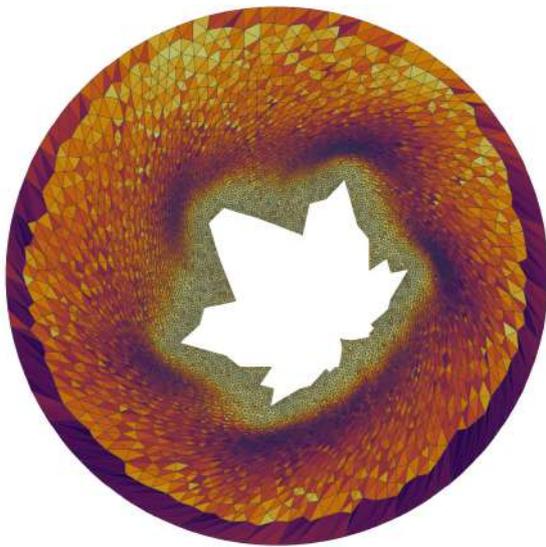
Table 5.7: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of sycamore leaf. Wendland C^2 with support radius $R_s = 11$ is the selected RBF kernel and Gaussian with $\varepsilon = 0.5$ is the weighting function.



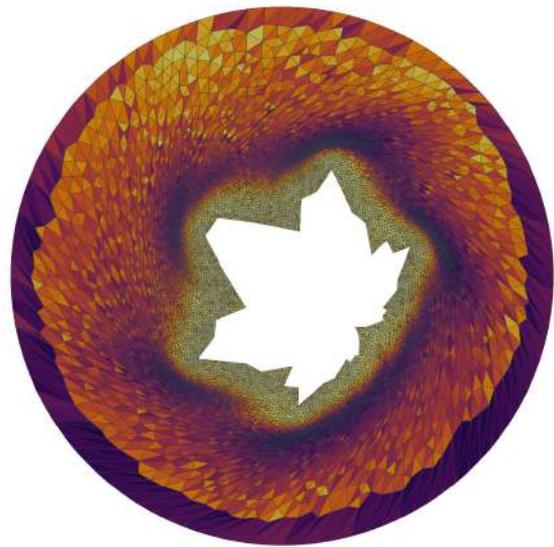
(a) 15 degrees angle of rotation



(b) 30 degrees angle of rotation



(c) 45 degrees angle of rotation



(d) 55 degrees angle of rotation



(e) Size-shape metric

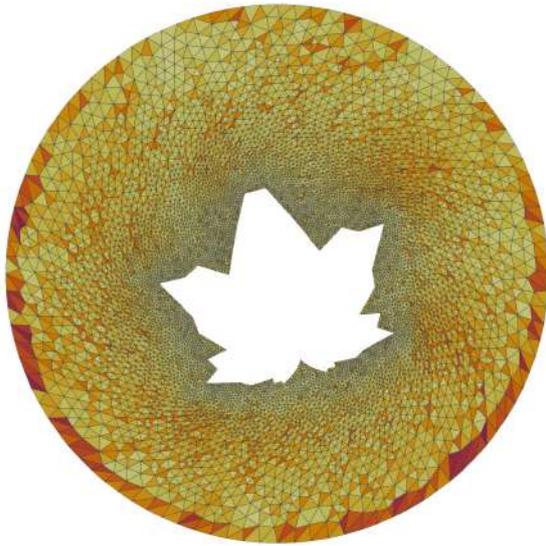
Figure 5.8: Application of the hybrid RBF mesh deformation method around a sycamore leaf. The leaf rotates 15, 30, 45 and 55 degrees around the origin. Wendland C^2 function with support radius $R_s = 11$ was chosen as the activation function of the RBF network, while Gaussian function with $\varepsilon = 0.5$ was the selected weighting function. Colour corresponds to size-shape metric f_{ss} . For angles of rotation greater than 58 degrees inverted elements appear.

5.2.8 Wendland C^2 ($R_s = 11$) with Gaussian ($\varepsilon = 1$) as weighting function

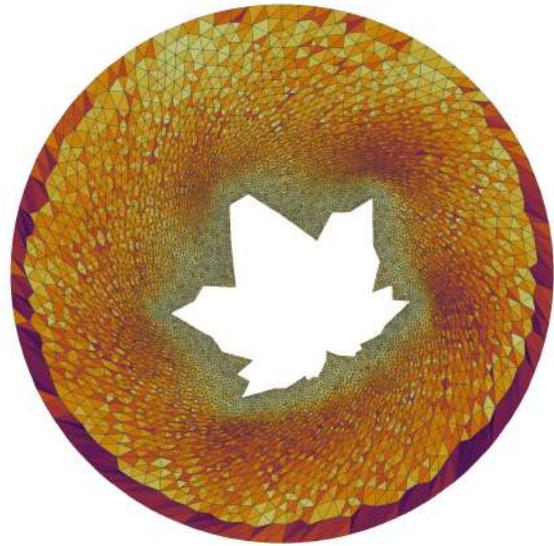
The proposed hybrid method is applied once more. Wendland C^2 function with a support radius $R_s = 11$ is used as the activation function of the RBF network. Gaussian function with a shape-parameter $\varepsilon = 1$ is used as the weighting function. In Table 5.8 the average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric for three angles of rotations are presented.

ϕ (degrees)	15	30	45	60
$\overline{f_{ss}}$	0.920	0.839	0.734	0.626
$\sigma_{f_{ss}}$	0.066	0.129	0.209	0.288
$f_{ss,\min}$	0.359	0.177	0.088	0.009

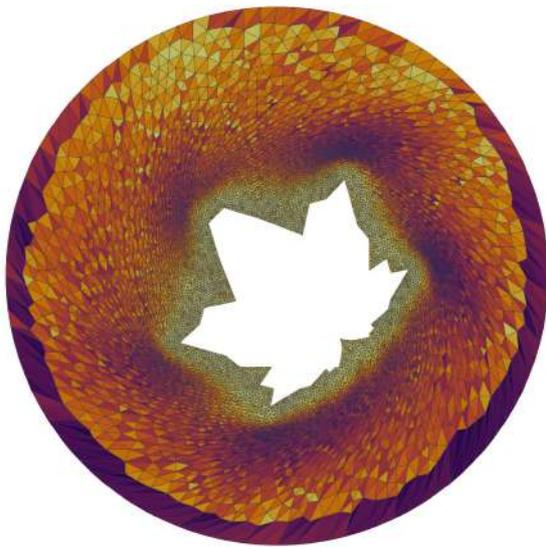
Table 5.8: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of sycamore leaf. Wendland C^2 with support radius $R_s = 11$ is the selected RBF kernel and Gaussian with $\varepsilon = 1$ is the weighting function.



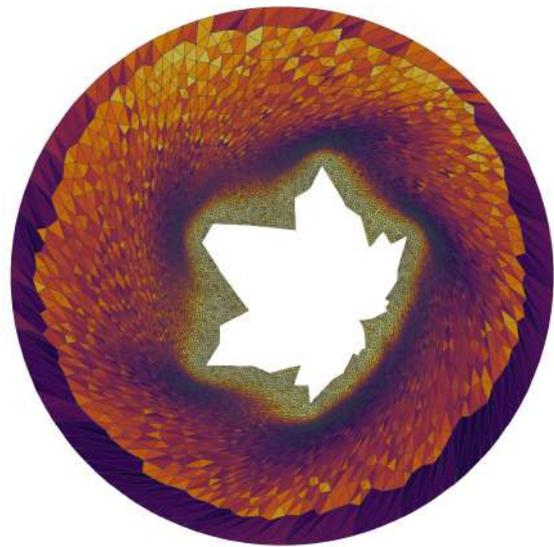
(a) 15 degrees angle of rotation



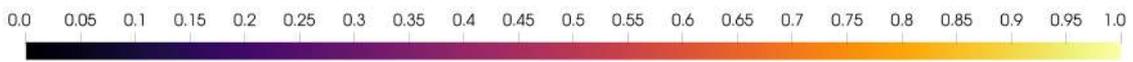
(b) 30 degrees angle of rotation



(c) 45 degrees angle of rotation



(d) 60 degrees angle of rotation



(e) Size-shape metric

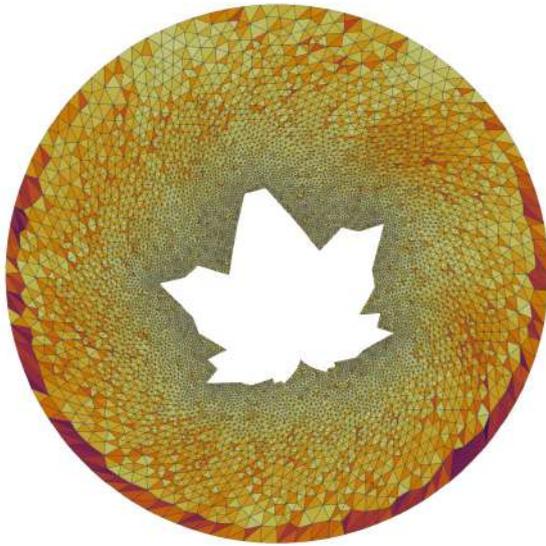
Figure 5.9: Application of the hybrid RBF mesh deformation method around a sycamore leaf. The leaf rotates 15, 30, 45 and 60 degrees around the origin. Wendland C^2 function with support radius $R_s = 11$ was chosen as the activation function of the RBF network, while Gaussian function with $\varepsilon = 1$ was the selected weighting function. Colour corresponds to size-shape metric f_{ss} . For angles of rotation greater than 64 degrees inverted elements appear.

5.2.9 Wendland C^2 ($R_s = 11$) with Wendland C^0 as weighting function

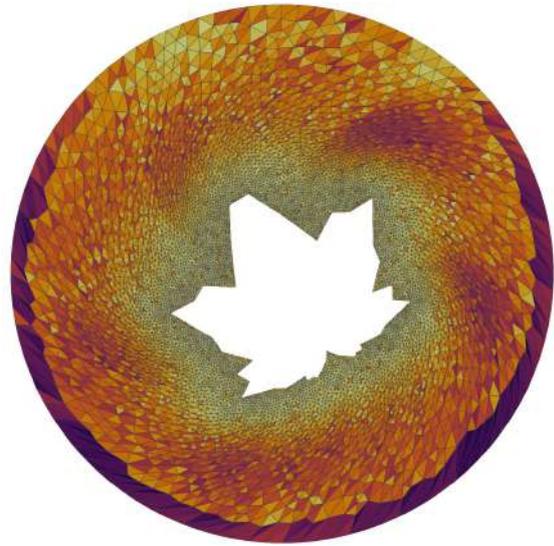
Lastly the proposed hybrid method is applied with Wendland C^0 function as the weighting function.. Wendland C^2 function with a support radius $R_s = 11$ is used as the activation function of the RBF network. In Table 5.9 the average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric for three angles of rotations are presented.

ϕ (degrees)	15	30	45	60
$\overline{f_{ss}}$	0.924	0.857	0.775	0.692
$\sigma_{f_{ss}}$	0.072	0.140	0.217	0.289
$f_{ss,\min}$	0.272	0.067	0.029	0.008

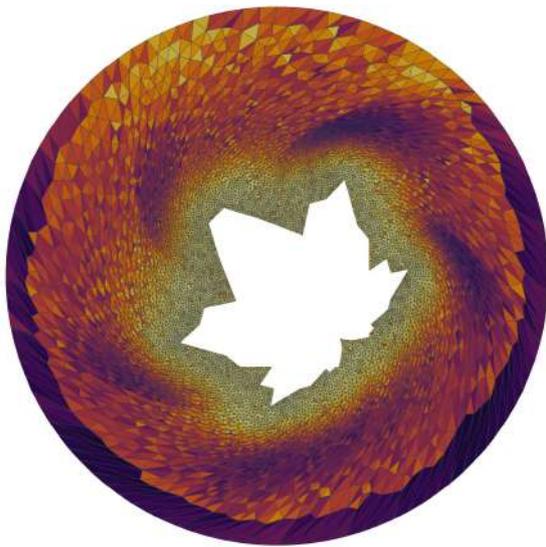
Table 5.9: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of sycamore leaf. Wendland C^2 with support radius $R_s = 11$ is the selected RBF kernel and Wendland C^0 is the weighting function.



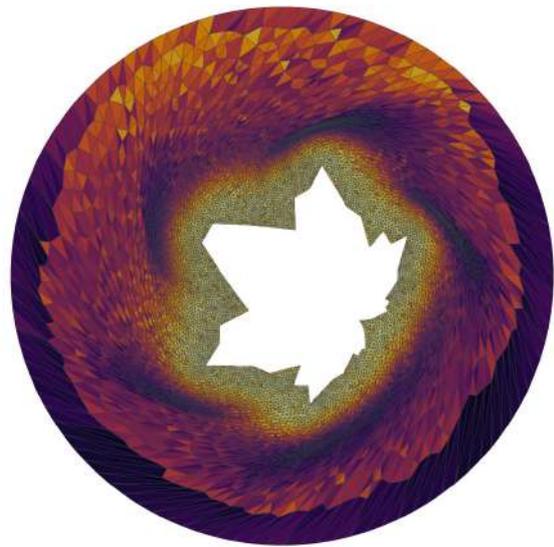
(a) 15 degrees angle of rotation



(b) 30 degrees angle of rotation



(c) 45 degrees angle of rotation



(d) 60 degrees angle of rotation



(e) Size-shape metric

Figure 5.10: Application of the hybrid RBF mesh deformation method around a sycamore leaf. The leaf rotates 15, 30, 45 and 60 degrees around the origin. Wendland C^2 function with support radius $R_s = 11$ was chosen as the activation function of the RBF network, while Wendland C^0 function, deployed as globally supported function, was the selected weighting function. Colour corresponds to size-shape metric f_{ss} . For angles of rotation greater than 61 degrees inverted elements appear.

5.3 Comments

Regarding the typical RBF mesh morphing method, it is observed that compactly supported functions, with a big enough radius of support (not big enough to enclose the whole mesh domain), have a more robust behavior than functions with global support. The deformed mesh is of higher quality for the same angle of rotation, while inverted elements appear in much greater angles. That is the case when the radius of support is big enough to influence all the internal mesh nodes. For smaller radius the deformed mesh presents inverted elements for considerably smaller angles of rotation. When Wendland C^2 , with $R_s = 11$, is used as RBF kernel the mesh can accommodate angles of rotation above sixty degrees. However, it should be noted that at these angles there are cells with f_{ss} near to zero. Even though the mesh does not present inverted elements, in practice it may be unsuitable for simulations since some cells are close to degeneration.

The hybrid RBF mesh morphing method achieves similar results when Gaussian with $\varepsilon = 1$ is used as weighting function. The deformed mesh in each angle of rotation does not present an improvement regarding the average size-shape metric compared to typical RBF method. The main asset of the hybrid method is the fact that near the boundary, where greater gradients of pressure and velocity occur, the mesh preserves its quality as it is shown in above figures. It is observed that cells with low quality appear farther from the boundary in comparison with typical RBF method.

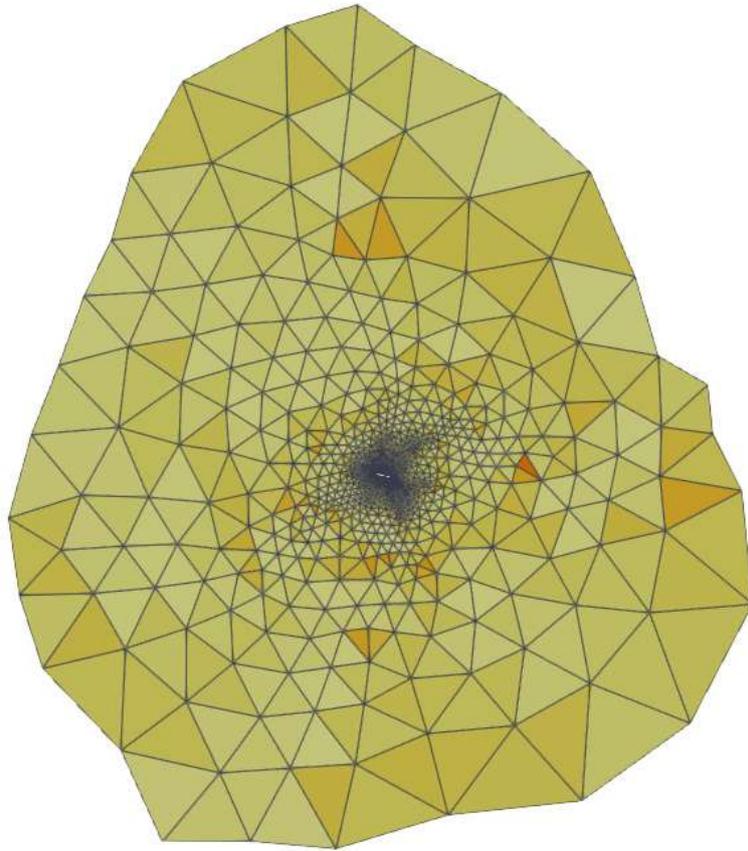
Chapter 6

Application to the mesh around a rotated four element airfoil

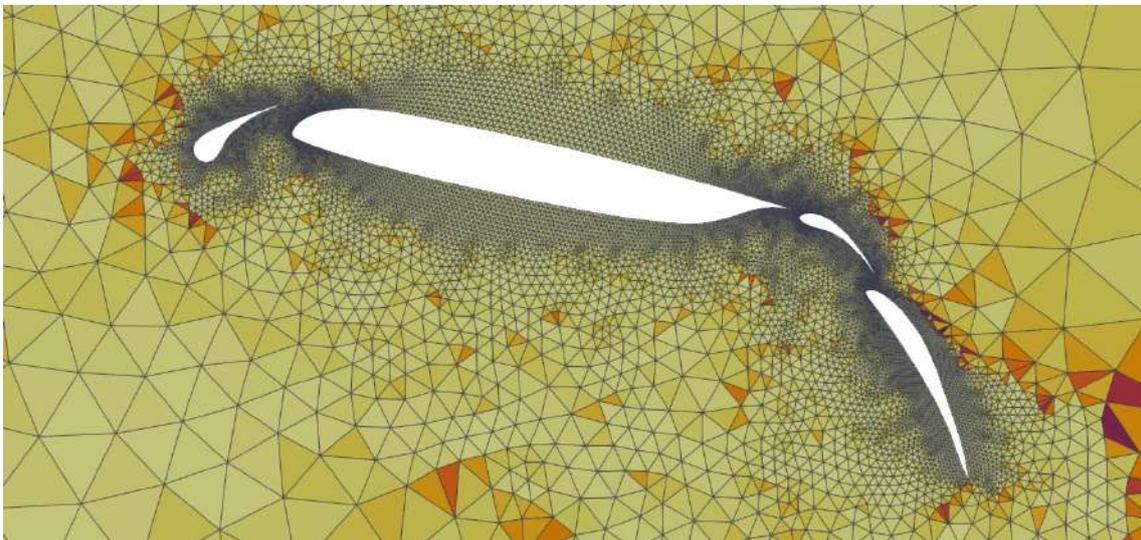
6.1 Problem description

In this chapter, the RBF method and the proposed hybrid method are applied to deform the mesh around a rotated four element airfoil within a stationary bounding contour. Between the airfoil and the outer contour, a mesh with triangular elements is generated using the advanced-front method [36] with a software of PCOpt/NTUA. The mesh is provided from PCOpt/NTUA in two files (i.e., LTT format: one file with nodes coordinates and one file with nodes connectivity), and it is then transformed in VTK file format in order to be compatible with ParaView [37]. For this purpose, software that converts both 2D and 3D LTT unstructured meshes into VTK format was developed. Additionally, every element of the mesh is colored according to its shape quality (see Appendix A) through a color-quality mapping. Figure 6.1 depicts the original mesh as it is visualized in ParaView.

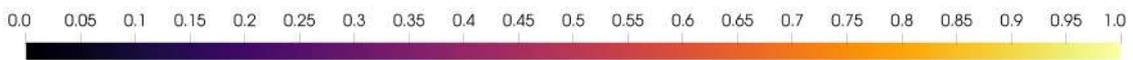
The airfoil rotates around the origin, and subsequently, the internal mesh is deformed, conforming to the updated boundary nodes locations. The aim is to test both methods with locally supported activation functions and compare the resulting deformed meshes.



(a) Whole mesh around the four-element airfoil



(b) Zoomed view of the mesh around the four-element airfoil



(c) Size-shape metric

Figure 6.1: The original mesh with triangular elements around a four-element airfoil as it was provided from software of PCOpt/NTUA

6.2 Results

At first the original RBF method will be tested using one activation function of global support and two activation functions of local support.

6.2.1 Exponential

In this section the globally supported exponential function (Equation 6.1) was chosen as activation function of the RBF network. The airfoil is repeatedly rotated in increasing angles until inverted elements appear.

$$e^{-\varepsilon d} \quad (6.1)$$

In order to understand the way that the decay of the kernel affects the resulted mesh two shape-parameters, ε , are utilized.

Shape parameter $\varepsilon = 0.1$

Here ε of Equation 6.1 is set equal to 0.1. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each angle of rotation are presented in Table 6.1. For angles of rotation greater than 60 degrees the resulted mesh presents inverted elements.

$\phi(r)$	15	30	45	60
$\overline{f_{ss}}$	0.960	0.930	0.883	0.822
$\sigma_{f_{ss}}$	0.048	0.068	0.102	0.150
$f_{ss,\min}$	0.088	0.075	0.060	0.045

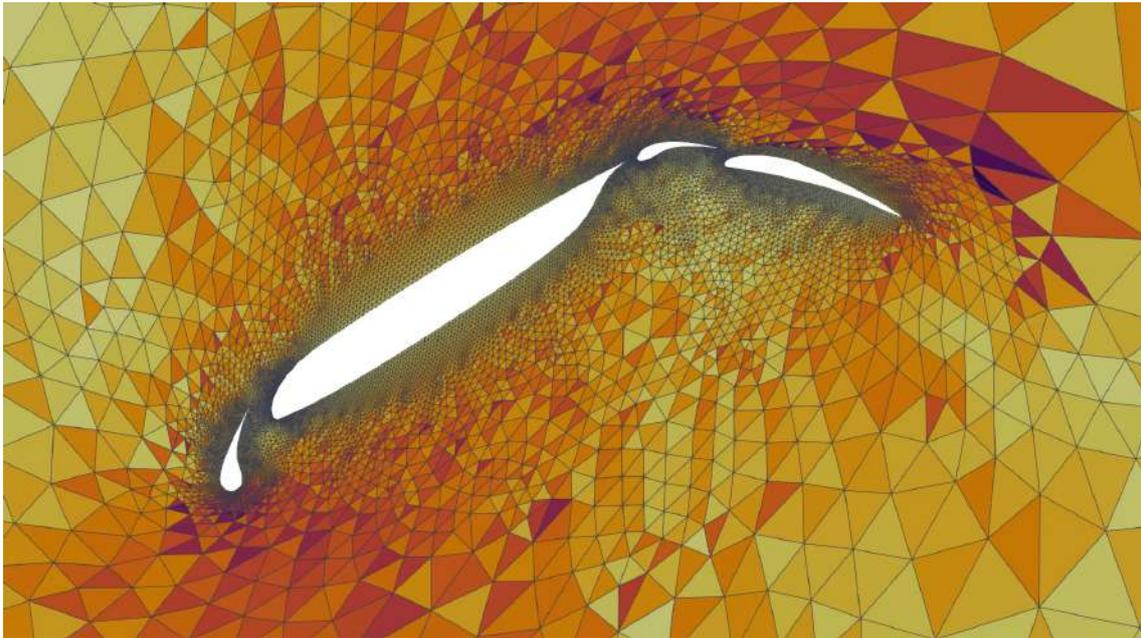
Table 6.1: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the airfoil. Exponential with $\varepsilon = 0.1$ is the selected RBF kernel.

Shape-parameter $\varepsilon = 1$

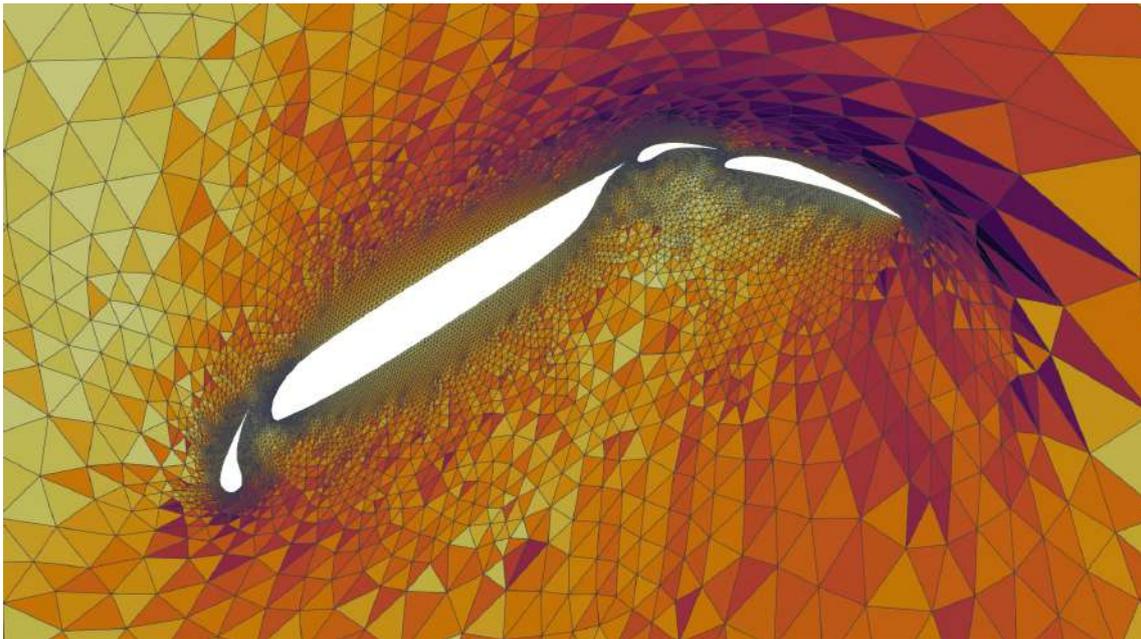
Next, ε of Equation 6.1 is set equal to 1. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric, that is introduced in Appendix A, for each angle of rotation are presented in Table 6.2. For angles of rotation greater than 55 degrees the resulted mesh presents inverted elements.

$\phi(\mathbf{r})$	15	30	45	55
$\overline{\mathbf{f}}_{\text{ss}}$	0.949	0.900	0.831	0.777
$\sigma_{\mathbf{f}_{\text{ss}}}$	0.057	0.099	0.160	0.205
$\mathbf{f}_{\text{ss},\text{min}}$	0.075	0.052	0.030	0.014

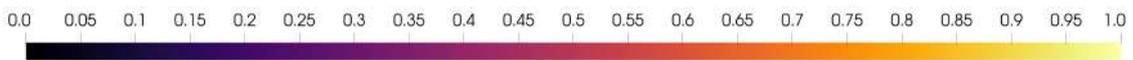
Table 6.2: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the airfoil. Exponential with $\varepsilon = 1$ is the selected RBF kernel.



(a) $\varepsilon = 0.1$



(b) $\varepsilon = 1$



(c) *Size-shape metric*

Figure 6.2: Application of the original RBF mesh deformation method around a four-element airfoil. The airfoil rotates 45 degrees around the origin. Exponential function with $\varepsilon = 0.1$ and $\varepsilon = 1$ was chosen as the activation function of the RBF network.

6.2.2 Wendland C^0

In this section the locally supported Wendland C^0 function (see Table 2.2) was chosen as activation function of the RBF network. The airfoil is repeatedly rotated in increasing angles until inverted elements appear. The activation function is deployed using two different radius of support, $R_s = 5$ and $R_s = 19$.

Support radius $R_s = 5$

Now, the support radius is increased to $R_s = 19$ in order to see if the deformed mesh is of higher quality. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each angle of rotation are presented in Table 6.3. For angles of rotation greater than 55 degrees the resulted mesh presents inverted elements.

$\phi(\mathbf{r})$	15	30	45	60
$\overline{f_{ss}}$	0.958	0.923	0.872	0.807
$\sigma_{f_{ss}}$	0.050	0.073	0.112	0.164
$f_{ss,\min}$	0.085	0.069	0.052	0.036

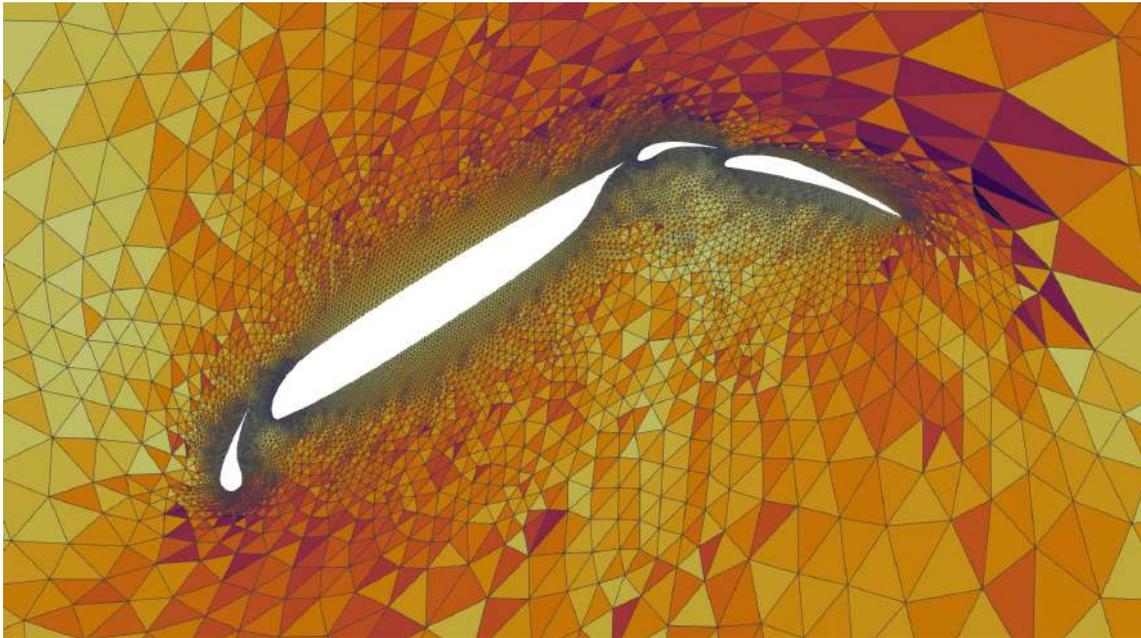
Table 6.3: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the airfoil. Wendland C^0 with support radius $R_s = 5$ is the selected RBF kernel.

Support radius $R_s = 19$

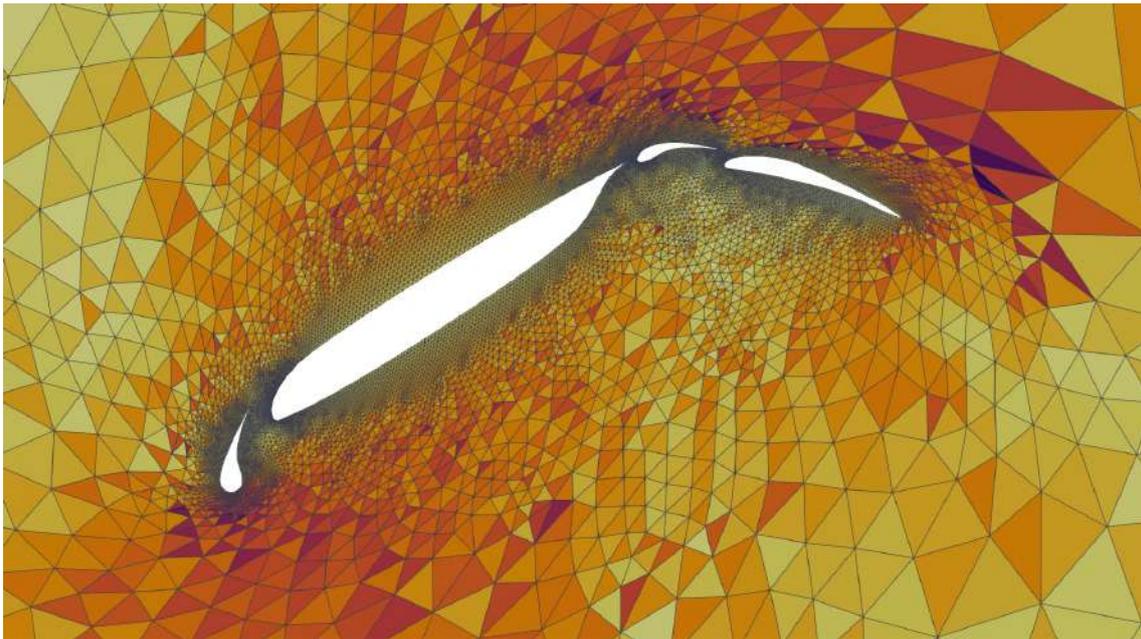
Now, the support radius is increased to $R_s = 19$ in order to see if the deformed mesh is of higher quality. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each angle of rotation are presented in Table 6.4. For angles of rotation greater than 55 degrees the resulted mesh presents inverted elements.

$\phi(\mathbf{r})$	15	30	45	60
$\overline{f_{ss}}$	0.961	0.931	0.885	0.825
$\sigma_{f_{ss}}$	0.048	0.067	0.101	0.147
$f_{ss,\min}$	0.088	0.075	0.061	0.046

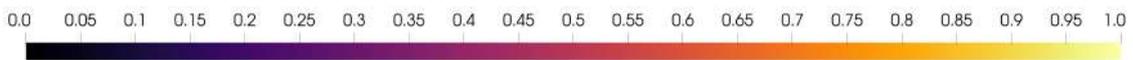
Table 6.4: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the airfoil. Wendland C^0 with support radius $R_s = 19$ is the selected RBF kernel.



(a) $R_s = 5$



(b) $R_s = 19$



(c) *Size-shape metric*

Figure 6.3: Application of the original RBF mesh deformation method around a four-element airfoil. The airfoil rotates 45 degrees around the origin. Wendland C^0 function with $R_s = 5$ and $R_s = 19$ was chosen as the activation function with compact support of the RBF network.

6.2.3 Wendland C^2

In this section the locally supported Wendland C^2 function (see Table 2.2) was chosen as activation function of the RBF network. The airfoil is repeatedly rotated in increasing angles until inverted elements appear. The activation function is deployed using two different radius of support, $R_s = 5$ and $R_s = 10$.

Support radius $R_s = 2$

The support radius is set to a relative small value, $R_s = 2$. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each angle of rotation are presented in Table 6.5. For angles of rotation greater than 50 degrees the resulted mesh presents inverted elements.

$\phi(\mathbf{r})$	15	30	45	50
$\overline{f_{ss}}$	0.967	0.954	0.939	0.934
$\sigma_{f_{ss}}$	0.049	0.068	0.093	0.103
$f_{ss,\min}$	0.093	0.088	0.042	0.005

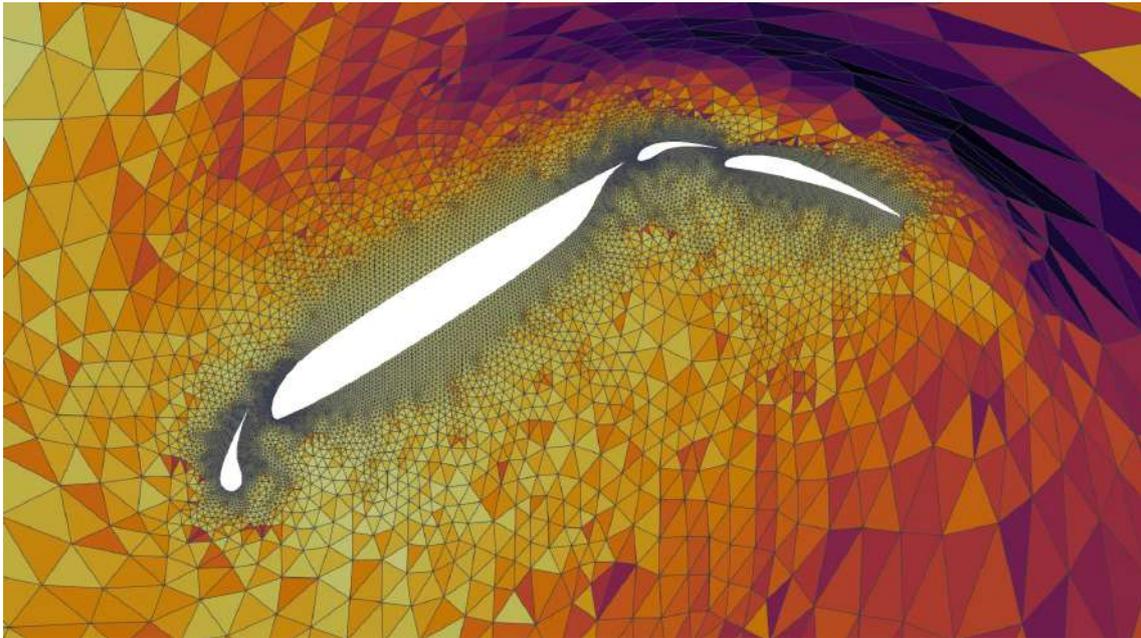
Table 6.5: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the airfoil. Wendland C^2 with support radius $R_s = 2$ is the selected RBF kernel.

Support radius $R_s = 10$

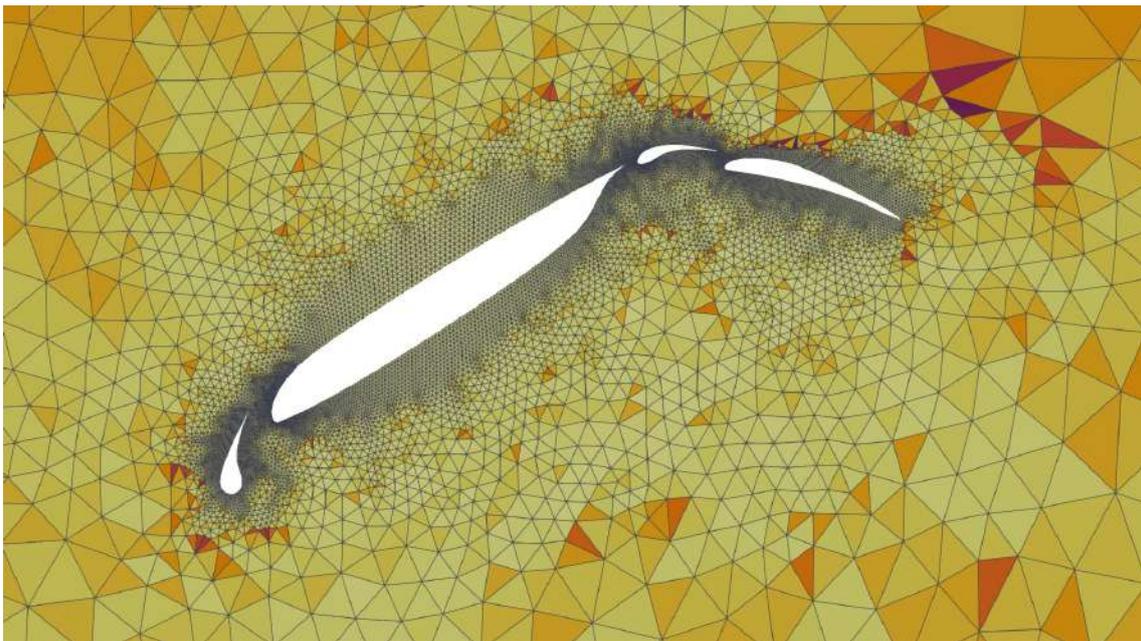
Now, the support radius is increased to $R_s = 10$ in order to see if the deformed mesh is of higher quality. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each angle of rotation are presented in Table 6.6. It is observed that Wendland C^2 with a big enough support radius (i.e. $R_s = 10$) can achieve deformed meshes with zero inverted elements even for angles of rotation equal to 90 degrees as it is depicted in Figure 6.5.

$\phi(\mathbf{r})$	15	30	45	60	75	90
$\overline{f_{ss}}$	0.976	0.974	0.971	0.966	0.961	0.956
$\sigma_{f_{ss}}$	0.043	0.045	0.052	0.063	0.078	0.097
$f_{ss,\min}$	0.098	0.098	0.097	0.097	0.096	0.096

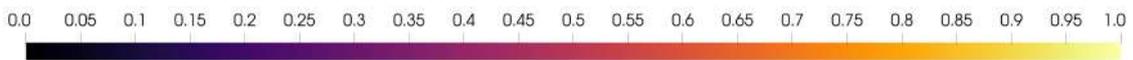
Table 6.6: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the airfoil. Wendland C^2 with support radius $R_s = 10$ is the selected RBF kernel.



(a) $R_s = 2$

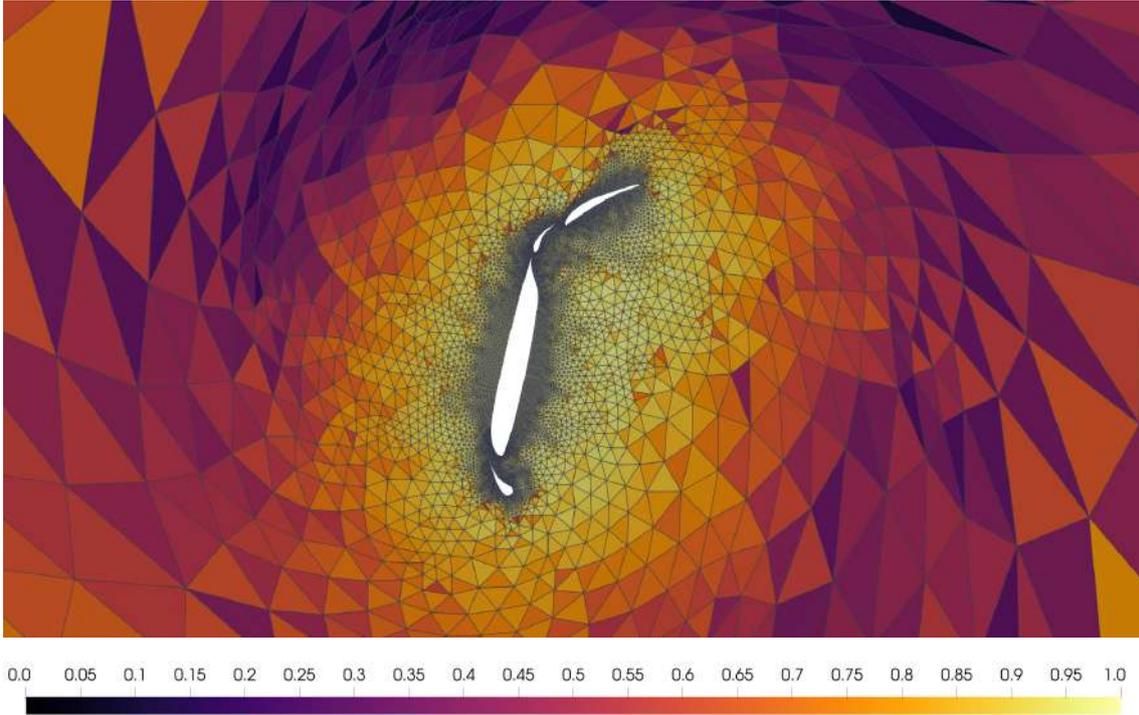


(b) $R_s = 10$



(c) *Size-shape metric*

Figure 6.4: Application of the original RBF mesh deformation method around a four-element airfoil. The airfoil rotates 45 degrees around the origin. Wendland C^2 function with $R_s = 2$ and $R_s = 10$ was chosen as the activation function with compact support of the RBF network.



(a) Size-shape metric

Figure 6.5: Application of the original RBF mesh deformation method around a four-element airfoil. The airfoil rotates 90 degrees around the origin. Wendland C^2 function with $R_s = 10$ was chosen as the activation function with compact support of the RBF network.

6.2.4 Wendland C^2 ($R_s = 2$) with Gaussian as weighting function

At this stage the proposed hybrid method is applied. Wendland C^2 function with a support radius $R_s = 2$ is used as the activation function of the RBF network. Gaussian function with $\varepsilon = 0.5$ and $\varepsilon = 2$ is used as the weighting function.

Shape-parameter of weighting function $\varepsilon = 0.5$

Shape-parameter of the weighting function is set equal to 0.5 in order to achieve a smooth transition. In Table 6.7 the average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric for four angles of rotation are presented.

$\phi(\mathbf{r})$	15	30	45	60
$\overline{f_{ss}}$	0.975	0.973	0.969	0.966
$\sigma_{f_{ss}}$	0.044	0.053	0.067	0.084
$f_{ss,\min}$	0.099	0.099	0.099	0.003

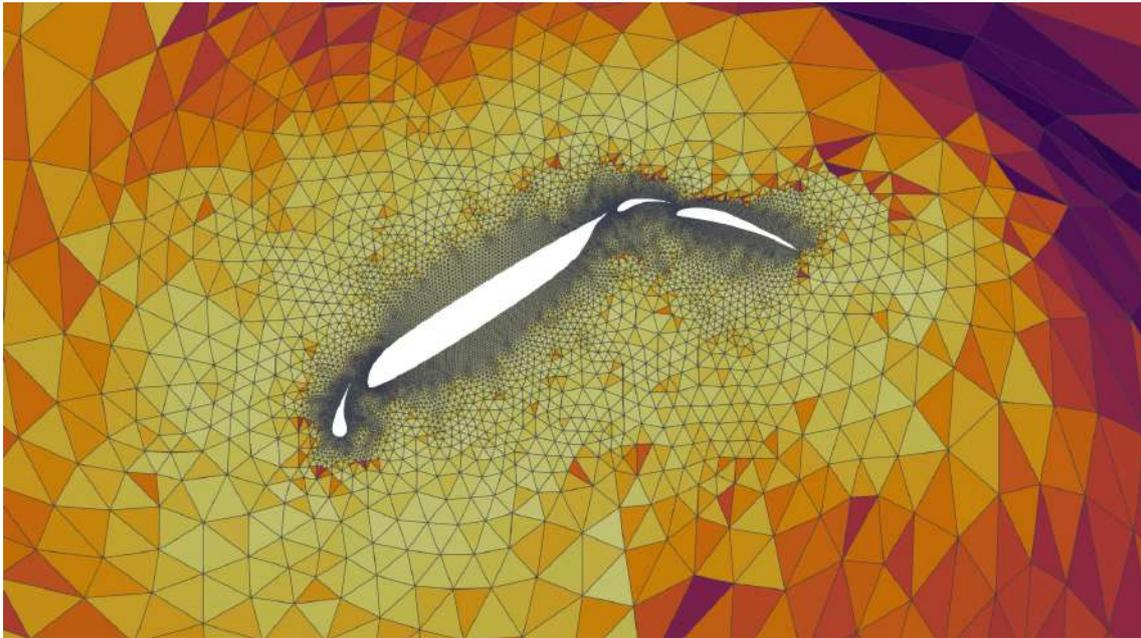
Table 6.7: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the airfoil. Wendland C^2 with support radius $R_s = 2$ is the selected RBF kernel and Gaussian function with shape-parameter $\varepsilon = 0.5$ is the weighting function.

Shape-parameter of weighting function $\varepsilon = 2$

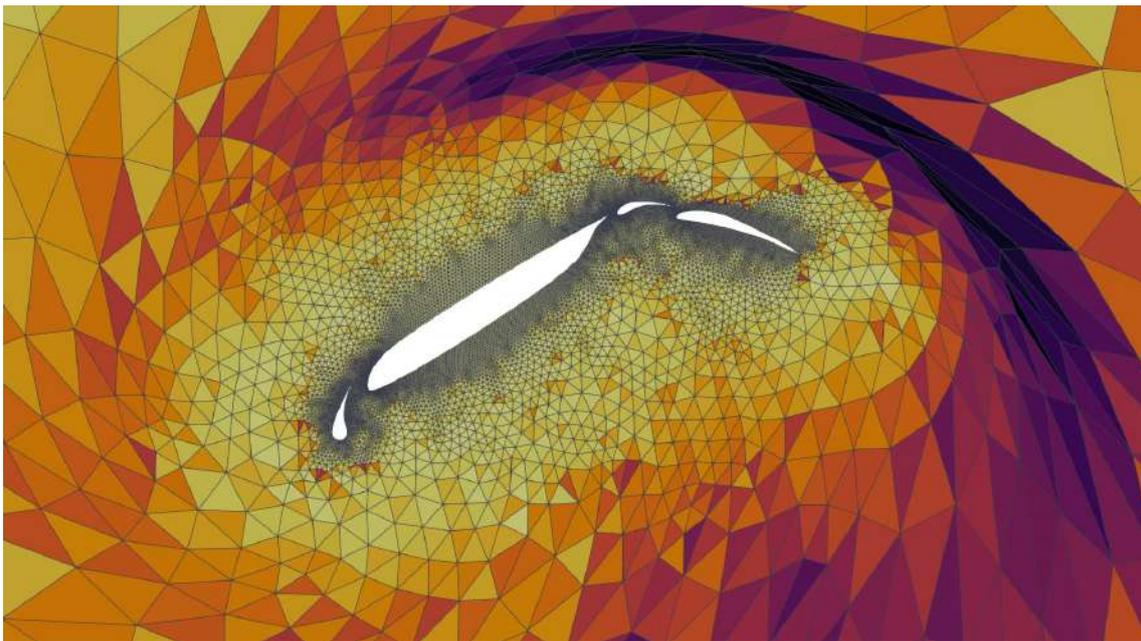
Shape-parameter of the weighting function is increased to 2 and thus the transition is less smooth. In Table 6.8 the average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric for three angles of rotation are presented.

$\phi(\mathbf{r})$	15	30	45
$\overline{f_{ss}}$	0.974	0.970	0.965
$\sigma_{f_{ss}}$	0.046	0.060	0.079
$f_{ss,\min}$	0.099	0.099	0.009

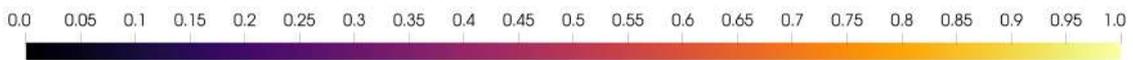
Table 6.8: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the airfoil. Wendland C^2 with support radius $R_s = 2$ is the selected RBF kernel and Gaussian function with shape-parameter $\varepsilon = 2$ is the weighting function.



(a) $\varepsilon = 0.5$



(b) $\varepsilon = 2$



(c) *Size-shape metric*

Figure 6.6: Application of the original RBF mesh deformation method around a four-element airfoil. The airfoil rotates 45 degrees around the origin. Wendland C^2 function with $R_s = 2$ was chosen as the activation function with compact support of the RBF network. Gaussian function with $\varepsilon = 0.5$ and $\varepsilon = 2$ is the selected weighting function.

6.2.5 Wendland C^2 ($R_s = 2$) with Wendland C^0 as weighting function

Following, the proposed hybrid method is applied using Wendland C^0 function as the weighting function, but deployed with global support. Similarly to previous test, Wendland C^2 with $R_s = 2$ is deployed again as the activation function of the RBF network. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each angle of rotation are presented in Table 6.9.

$\phi(\mathbf{r})$	15	30	45	60
$\overline{f_{ss}}$	0.976	0.975	0.973	0.970
$\sigma_{f_{ss}}$	0.043	0.044	0.048	0.055
$f_{ss,\min}$	0.099	0.099	0.099	0.099

Table 6.9: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the airfoil. Wendland C^2 with support radius $R_s = 2$ is the selected RBF kernel and Wendland C^0 function deployed with global support is the weighting function.

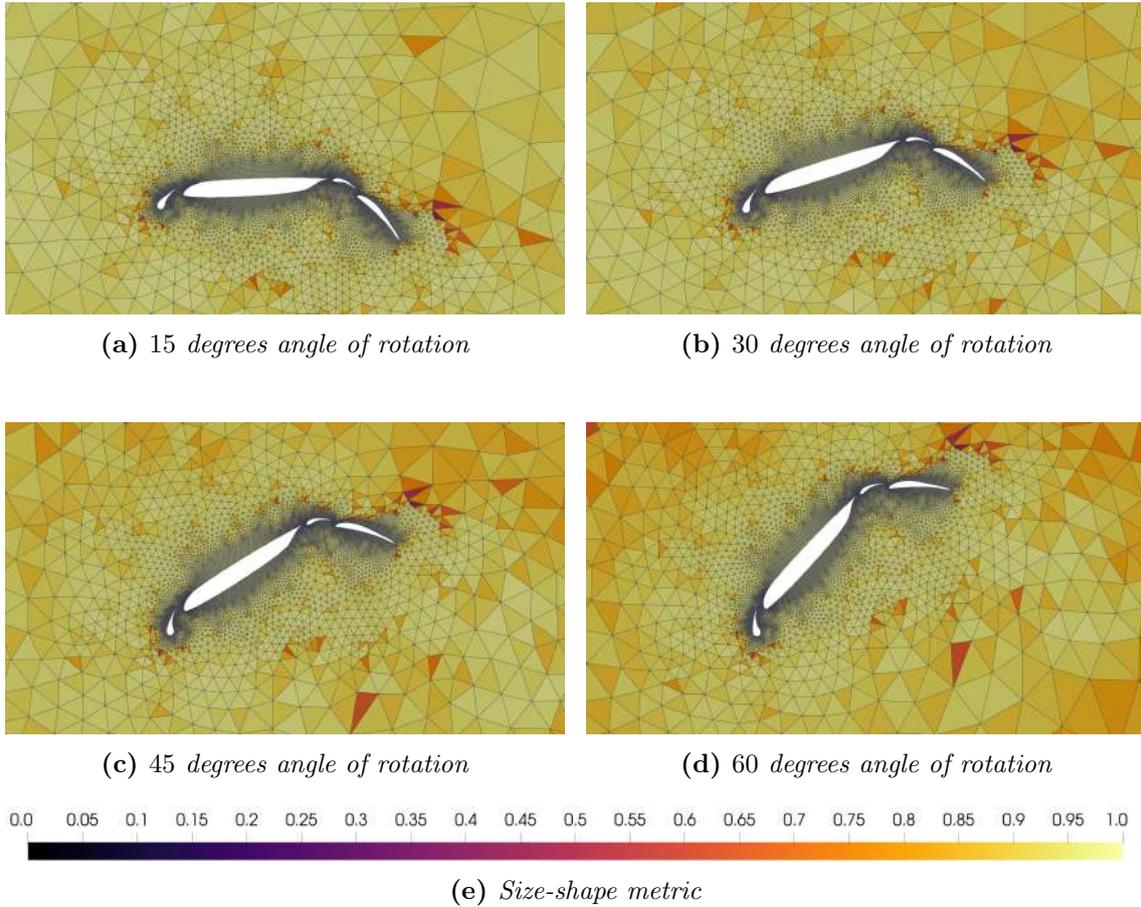


Figure 6.7: Application of the hybrid RBF mesh deformation method around a four element airfoil. The airfoil rotates 15, 30, 45 and 60 degrees around the origin. Wendland C^2 function with support radius $R_s = 2$ was chosen as the activation function of the RBF network, while Wendland C^0 function with deployed with global support was the selected weighting function. Colour corresponds to size-shape metric f_{ss} .

6.3 Comments

The mesh around the airfoil has the peculiarity that it is quite sparse away from it. Using an appropriate activation function the mesh does not present inverted elements even for extremely large angles of rotation. For that reason exponential function with smaller shape-parameter, ε (smoother decay), achieves better results. Similarly, compactly supported Wendland C^0 and Wendland C^2 produce meshes of higher quality as radius of support R_s increases. For instance, when Wendland C^2 with $R_s = 10$ is utilized as the activation function of the RBF network the

method can support angles of rotation greater than 90 degrees. Even though such big angles are not often encountered in an optimization cycle, that is not the case for aeroelasticity problems or flow simulations in the presence of moving bodies.

The hybrid RBF mesh morphing method achieves similar results when Wendland C^0 is used as weighting function. The deformed mesh in each angle of rotation does not present an improvement regarding the average size-shape metric compared to typical RBF method. At this case, the accomplishment of the hybrid method is the fact that can produce deformed meshes of good enough quality but using relative small support radius for the RBF kernel.

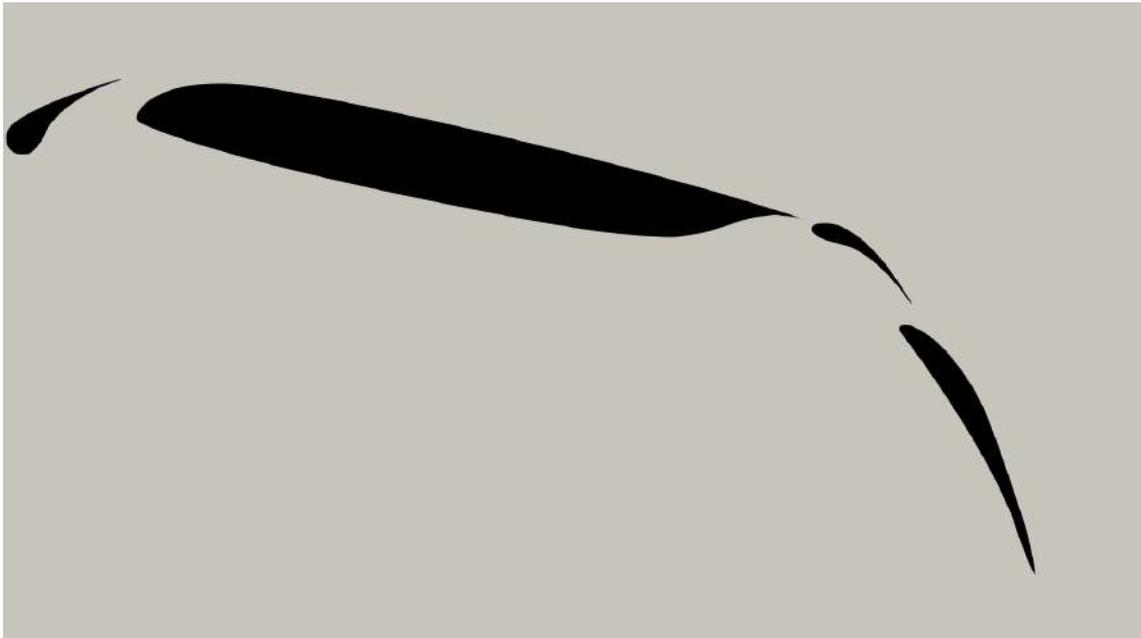
Chapter 7

Application to the mesh around a four element airfoil with moving slats and flaps

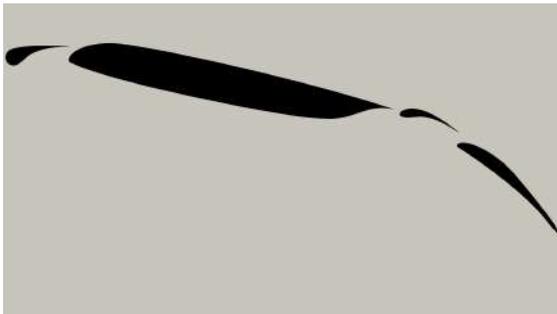
7.1 Problem description

In this chapter, the RBF method and the proposed hybrid method are applied to deform the mesh around a four element airfoil, which contains moving slats and flaps, within a stationary bounding contour. Between the airfoil and the outer contour, a mesh with triangular elements is generated using the advanced-front method [36] with a software of PCOpt/NTUA. The mesh is provided from PCOpt/NTUA in two files (i.e., LTT format: one file with nodes coordinates and one file with nodes connectivity), and it is then transformed in VTK file format in order to be compatible with ParaView [37]. For this purpose, software that converts both 2D and 3D LTT unstructured meshes into VTK format was developed. Additionally, every element of the mesh is colored according to its shape quality (see Appendix A) through a color-quality mapping. Figure 6.1 depicts the original mesh as it is visualized in ParaView.

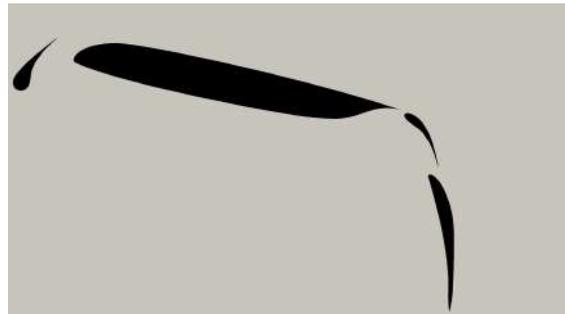
The flaps and the slats of the airfoil move up or down while the main element of the airfoil remains stationary. An instance of this movement is depicted in Figure 7.1. Subsequently, the internal mesh is deformed, conforming to the updated boundary nodes locations. The aim is to test both methods with locally supported activation functions and compare the resulting deformed meshes.



(a) *Initial four element airfoil configuration*



(b) *Slats and flaps move upwards*



(c) *Slats and flaps move downwards*

Figure 7.1: *Slats and flaps movement*

7.2 Results

At first the original RBF method will be tested using one activation function of global support and two activation functions of local support.

7.2.1 Exponential

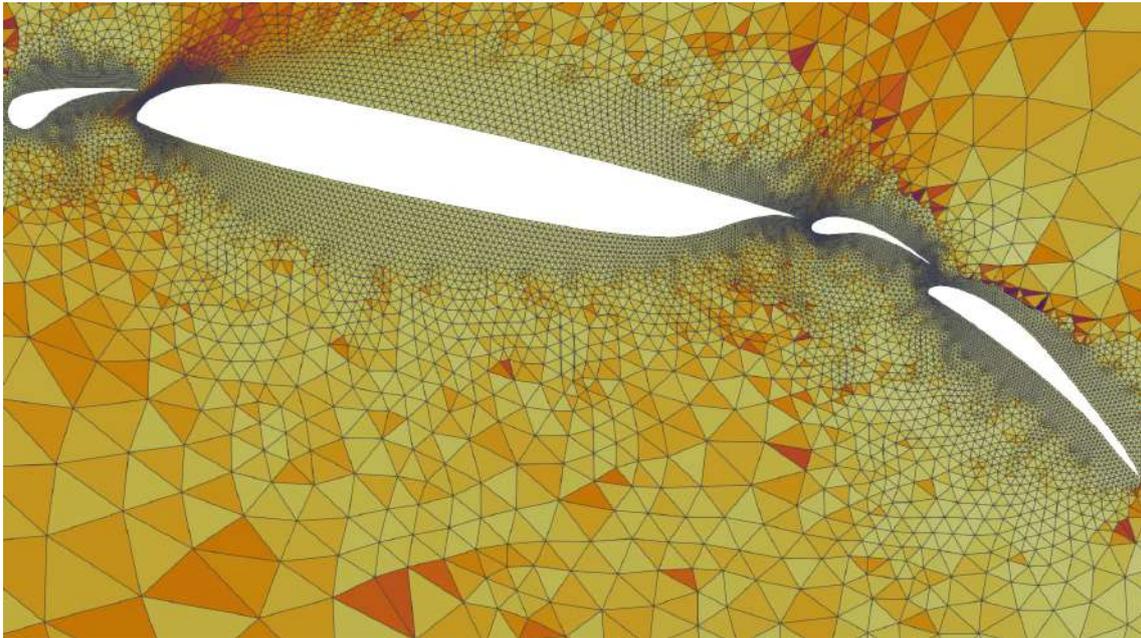
In this section the globally supported exponential function (Equation 6.1) was chosen as activation function of the RBF network. Slats and flaps of the airfoil move as described above.

Shape parameter $\varepsilon = 0.1$

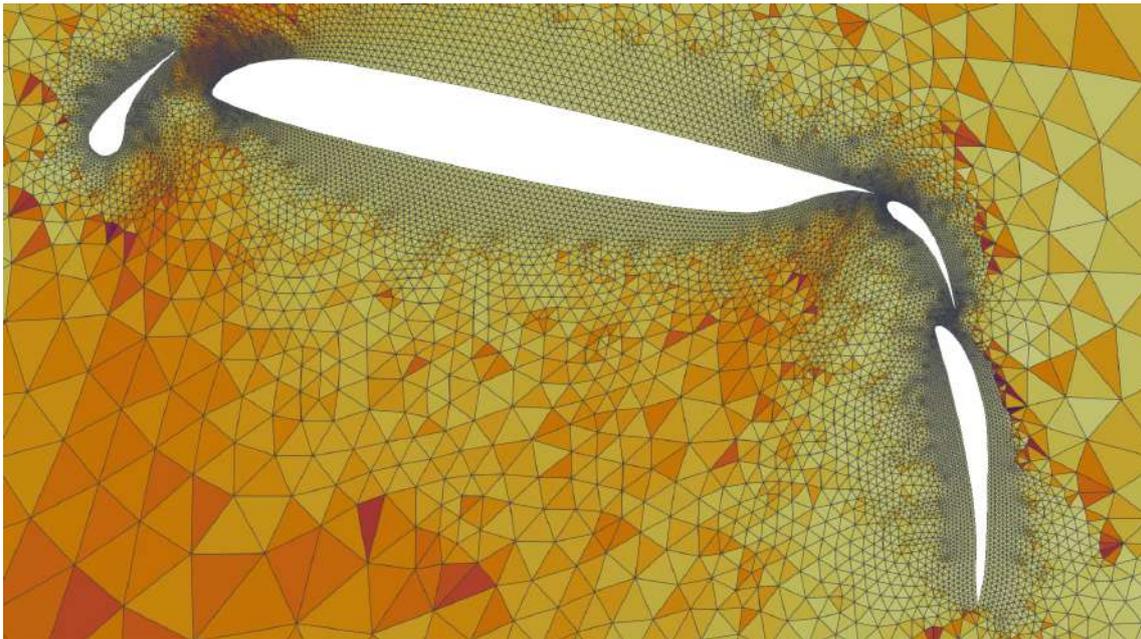
Here ε of Equation 6.1 is set equal to 0.1. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each movement of the elements of the airfoil are presented in Table 7.1.

movement	upwards	downwards
$\overline{f_{ss}}$	0.923	0.933
$\sigma_{f_{ss}}$	0.094	0.066
$f_{ss,\min}$	0.106	0.106

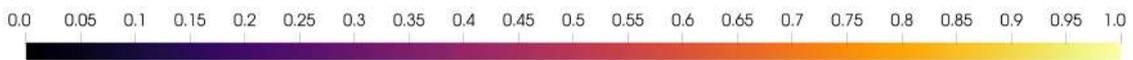
Table 7.1: Average value, standard deviation and minimum value of the mesh quality metric for each movement of airfoil's elements. Exponential with $\varepsilon = 0.1$ is the selected RBF kernel.



(a) *Slats and flaps move upwards*



(b) *Slats and flaps move downwards*



(c) *Size-shape metric*

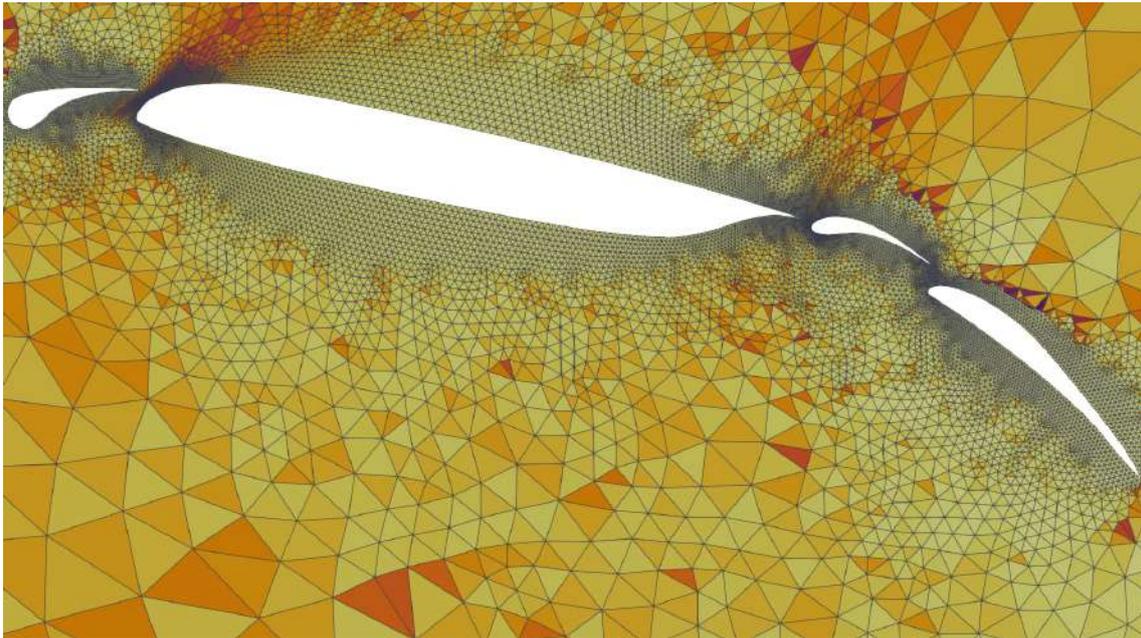
Figure 7.2: Application of the original RBF mesh deformation method around a four-element airfoil. Slats and flaps of the airfoil move upwards and downwards. Exponential function with $\epsilon = 0.1$ was chosen as the activation function of the RBF network.

Shape parameter $\varepsilon = 2$

Following, ε of Equation 6.1 is increased to 2 and thus the RBF decays more sharply. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each movement of the elements of the airfoil are presented in Table 7.2.

movement	upwards	downwards
$\overline{f_{ss}}$	0.901	0.924
$\sigma_{f_{ss}}$	0.147	0.085
$f_{ss,\min}$	0.099	0.098

Table 7.2: Average value, standard deviation and minimum value of the mesh quality metric for each movement of airfoil's elements. Exponential with $\varepsilon = 2$ is the selected RBF kernel.



(a) *Slats and flaps move upwards*



(b) *Slats and flaps move downwards*



(c) *Size-shape metric*

Figure 7.3: Application of the original RBF mesh deformation method around a four-element airfoil. Slats and flaps of the airfoil move upwards and downwards. Exponential function with $\varepsilon = 2$ was chosen as the activation function of the RBF network.

7.2.2 Wendland C^0

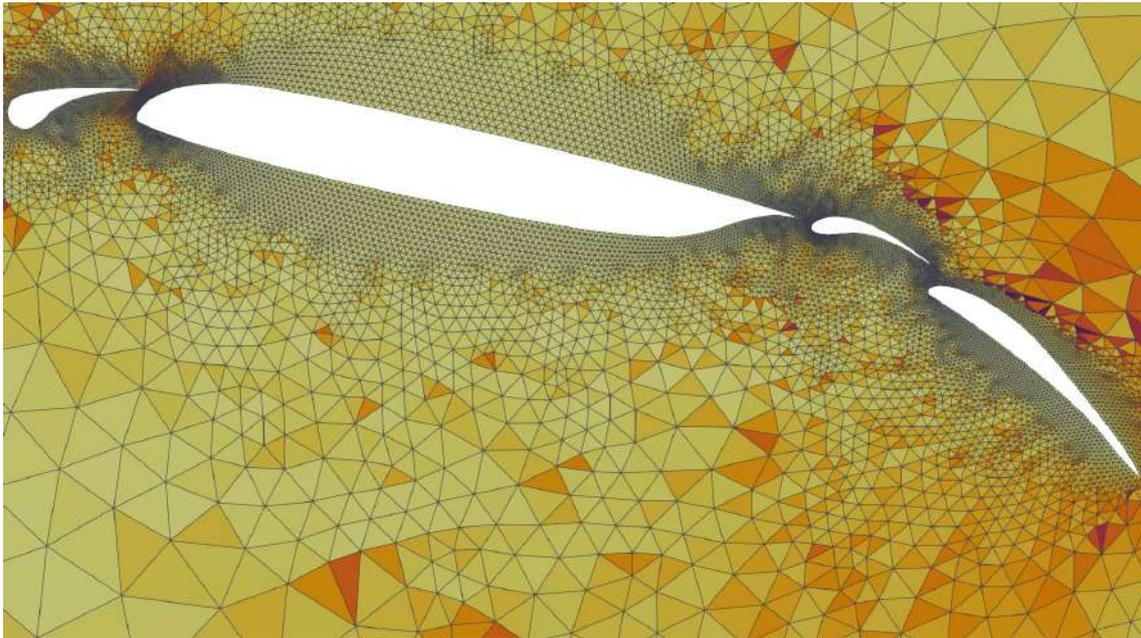
In this section the locally supported Wendland C^0 function (see Table 2.2) was chosen as activation function of the RBF network. Slats and flaps of the airfoil move as described above.

Radius of support $R_s = 1$

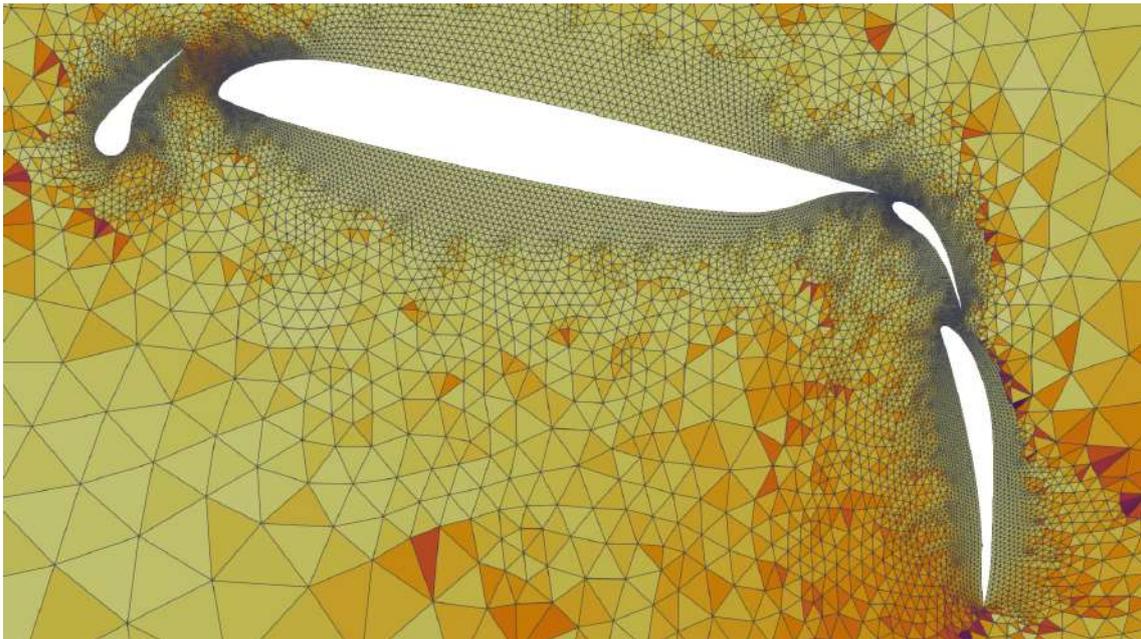
Here radius of support of the RBF kernel is set equal to 1. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each movement of the elements of the airfoil are presented in Table 7.3.

movement	upwards	downwards
$\overline{f_{ss}}$	0.917	0.932
$\sigma_{f_{ss}}$	0.098	0.070
$f_{ss,\min}$	0.071	0.115

Table 7.3: Average value, standard deviation and minimum value of the mesh quality metric for each movement of airfoil's elements. Wendland C^0 with $R_s = 1$ is the selected RBF kernel.



(a) *Slats and flaps move upwards*



(b) *Slats and flaps move downwards*



(c) *Size-shape metric*

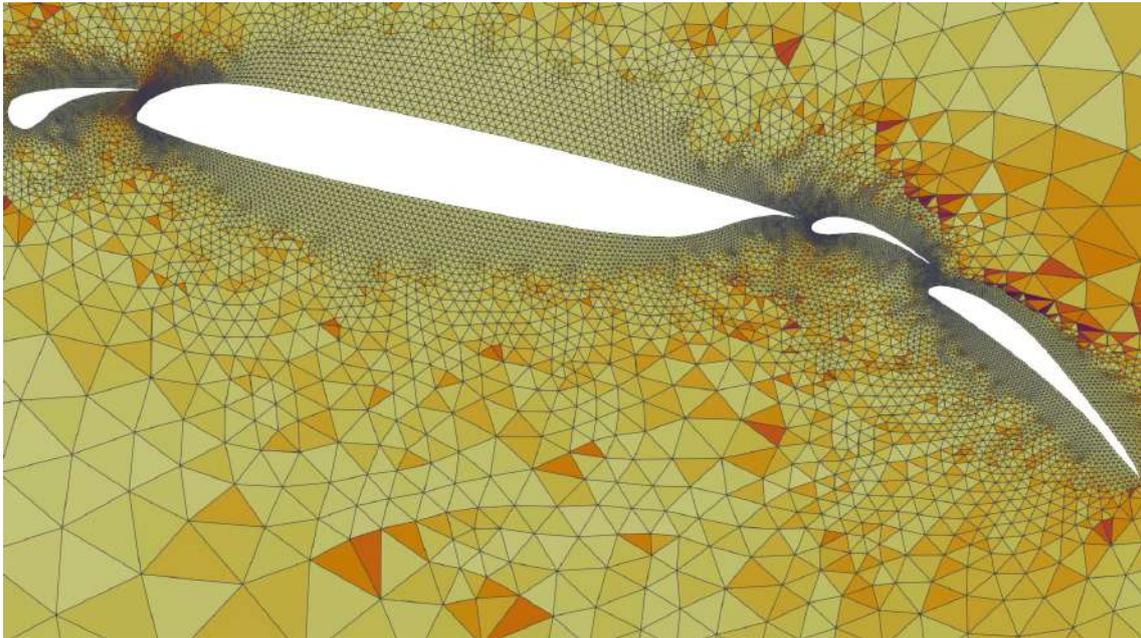
Figure 7.4: Application of the original RBF mesh deformation method around a four-element airfoil. Slats and flaps of the airfoil move upwards and downwards. Wendland C^0 with $R_s = 1$ was chosen as the activation function of the RBF network.

Radius of support $R_s = 5$

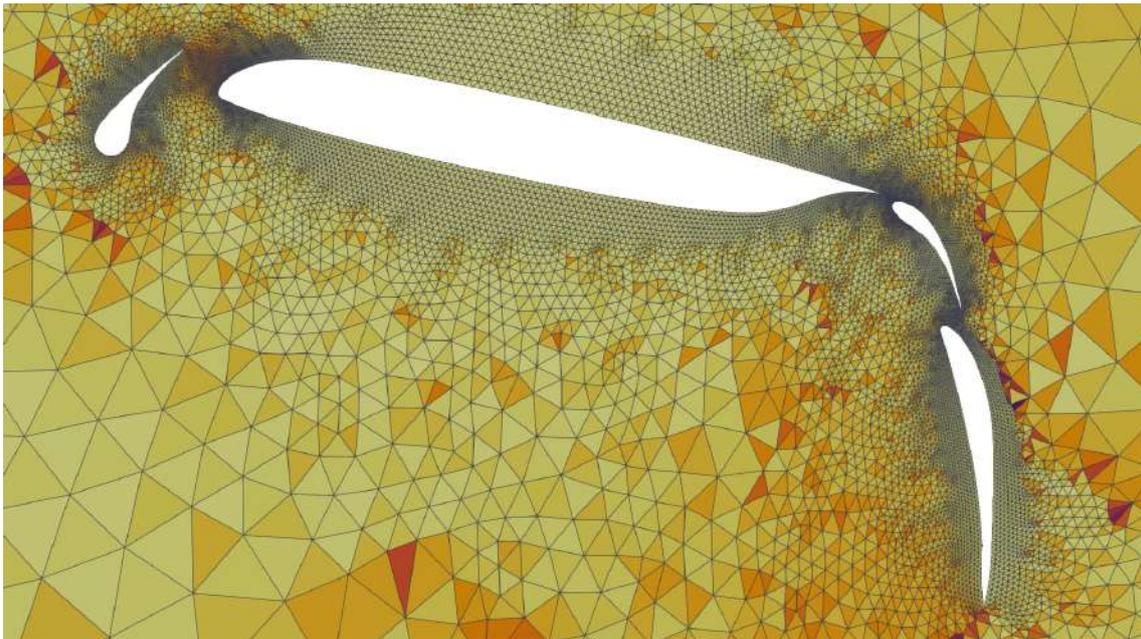
Here radius of support of the RBF kernel is increased to 5. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each movement of the elements of the airfoil are presented in Table 7.4.

movement	upwards	downwards
$\overline{f_{ss}}$	0.921	0.934
$\sigma_{f_{ss}}$	0.095	0.067
$f_{ss,\min}$	0.081	0.108

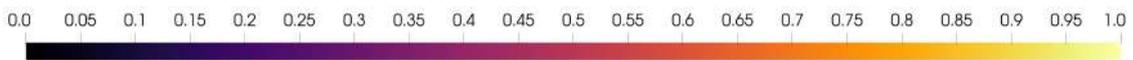
Table 7.4: Average value, standard deviation and minimum value of the mesh quality metric for each movement of airfoil's elements. Wendland C^0 with $R_s = 5$ is the selected RBF kernel.



(a) *Slats and flaps move upwards*



(b) *Slats and flaps move downwards*



(c) *Size-shape metric*

Figure 7.5: *Application of the original RBF mesh deformation method around a four-element airfoil. Slats and flaps of the airfoil move upwards and downwards. Wendland C^0 with $R_s = 5$ was chosen as the activation function of the RBF network.*

7.2.3 Wendland C^2

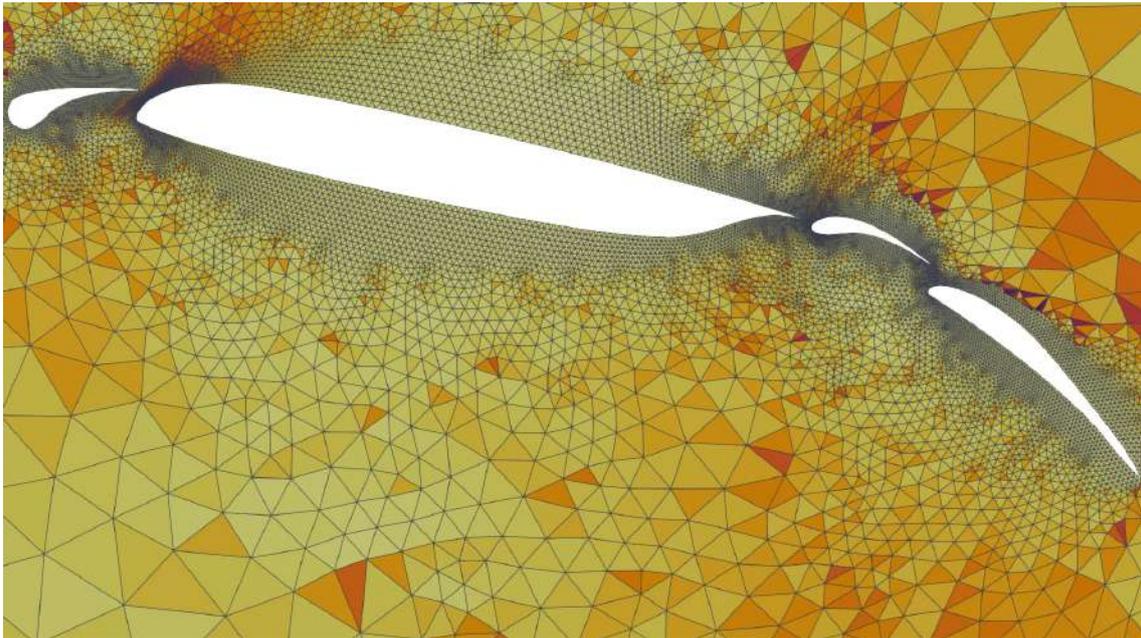
Lastly, the locally supported Wendland C^2 function (see Table 2.2) was chosen as activation function of the RBF network. Slats and flaps of the airfoil move as described above.

Radius of support $R_s = 1$

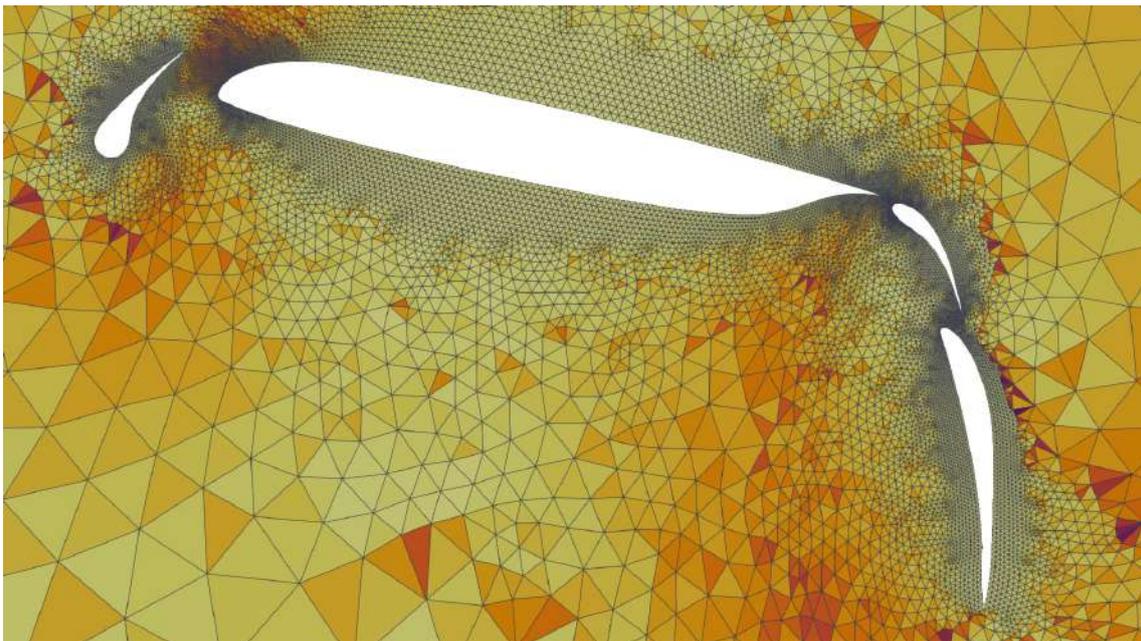
Here radius of support of the RBF kernel is set equal to 1. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each movement of the elements of the airfoil are presented in Table 7.5.

movement	upwards	downwards
$\overline{f_{ss}}$	0.904	0.928
$\sigma_{f_{ss}}$	0.141	0.084
$f_{ss,\min}$	0.090	0.103

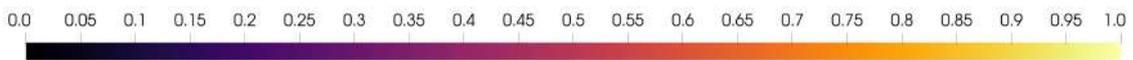
Table 7.5: Average value, standard deviation and minimum value of the mesh quality metric for each movement of airfoil's elements. Wendland C^2 with $R_s = 1$ is the selected RBF kernel.



(a) *Slats and flaps move upwards*



(b) *Slats and flaps move downwards*



(c) *Size-shape metric*

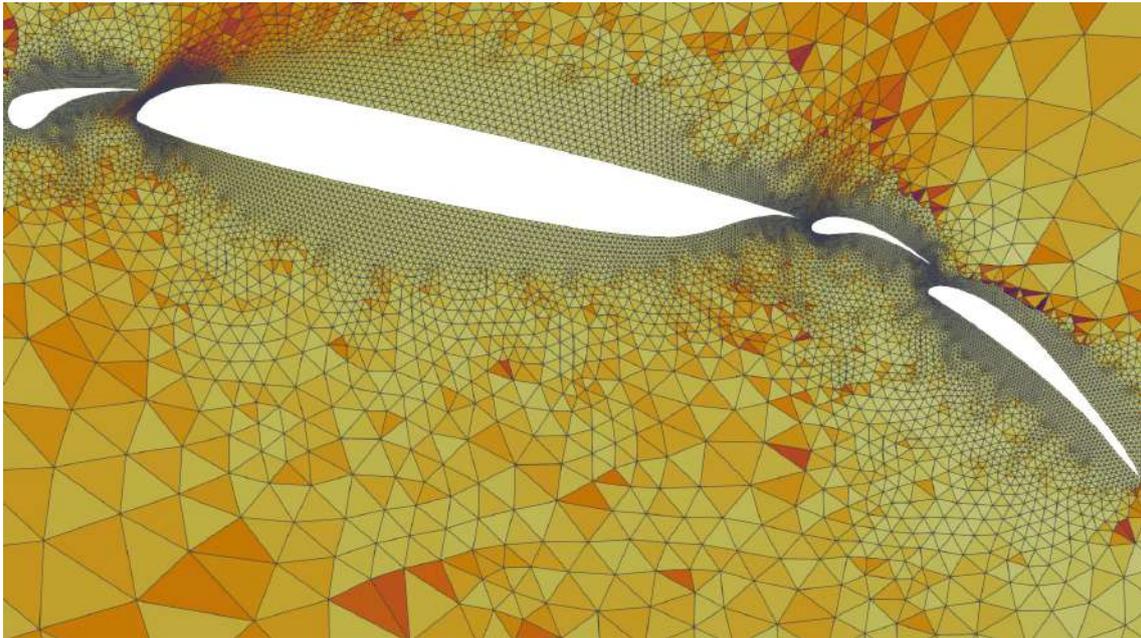
Figure 7.6: *Application of the original RBF mesh deformation method around a four-element airfoil. Slats and flaps of the airfoil move upwards and downwards. Wendland C^2 with $R_s = 1$ was chosen as the activation function of the RBF network.*

Radius of support $R_s = 5$

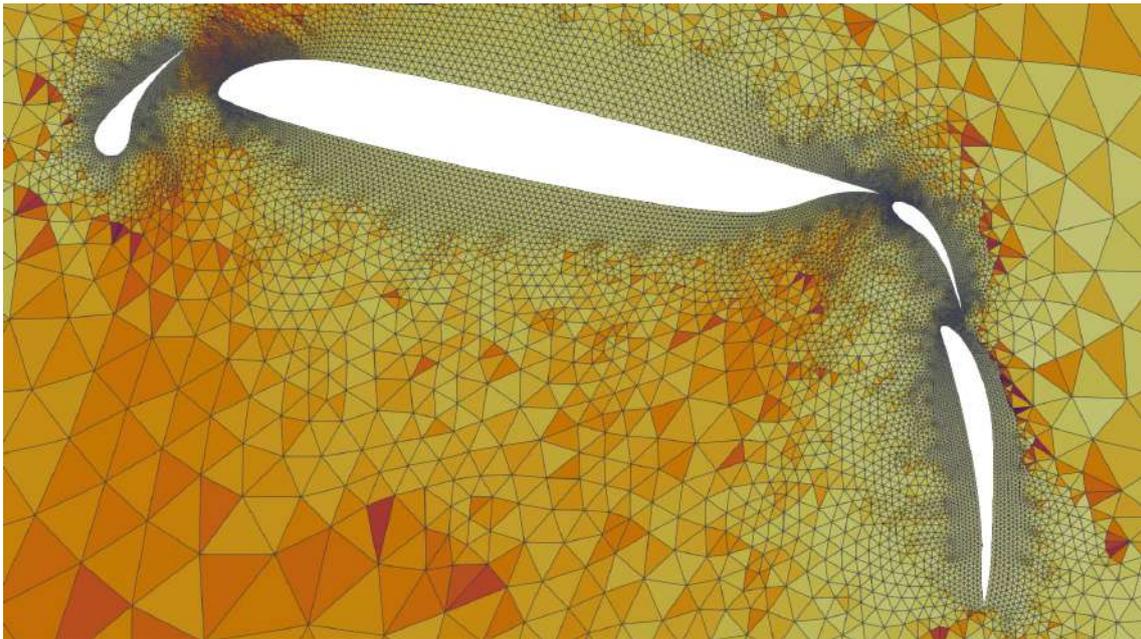
Here radius of support of the RBF kernel is increased to 5. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for each movement of the elements of the airfoil are presented in Table 7.6.

movement	upwards	downwards
$\overline{f_{ss}}$	0.903	0.925
$\sigma_{f_{ss}}$	0.143	0.084
$f_{ss,\min}$	0.099	0.099

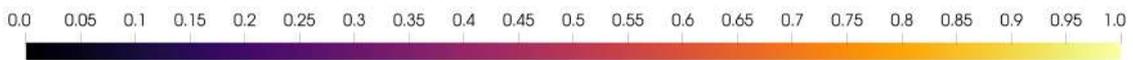
Table 7.6: Average value, standard deviation and minimum value of the mesh quality metric for each movement of airfoil's elements. Wendland C^2 with $R_s = 5$ is the selected RBF kernel.



(a) *Slats and flaps move upwards*



(b) *Slats and flaps move downwards*

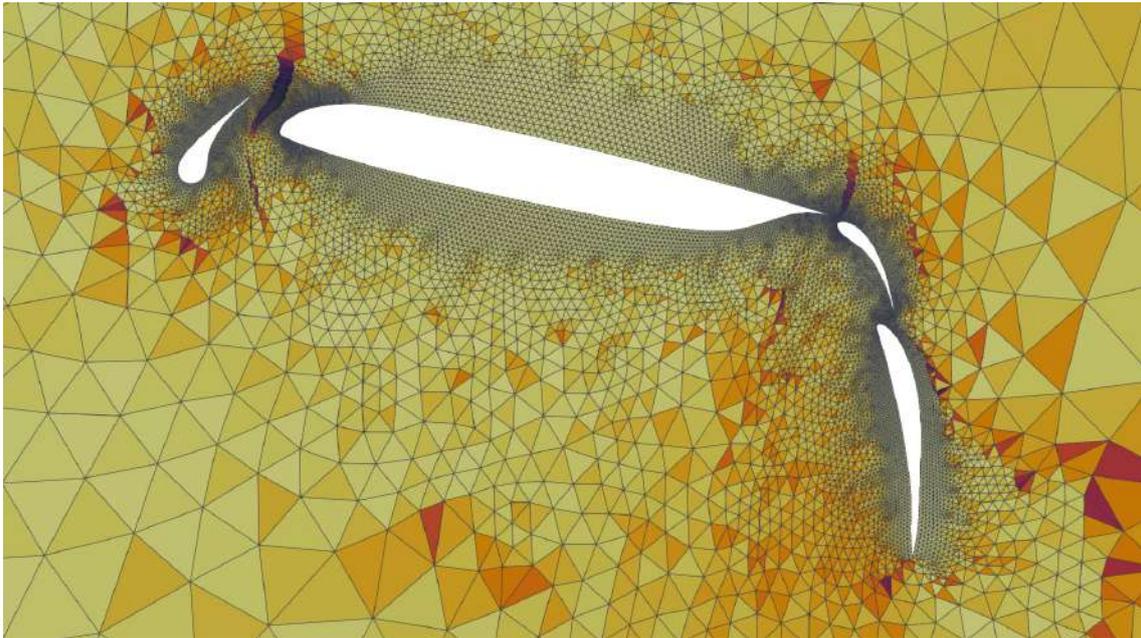


(c) *Size-shape metric*

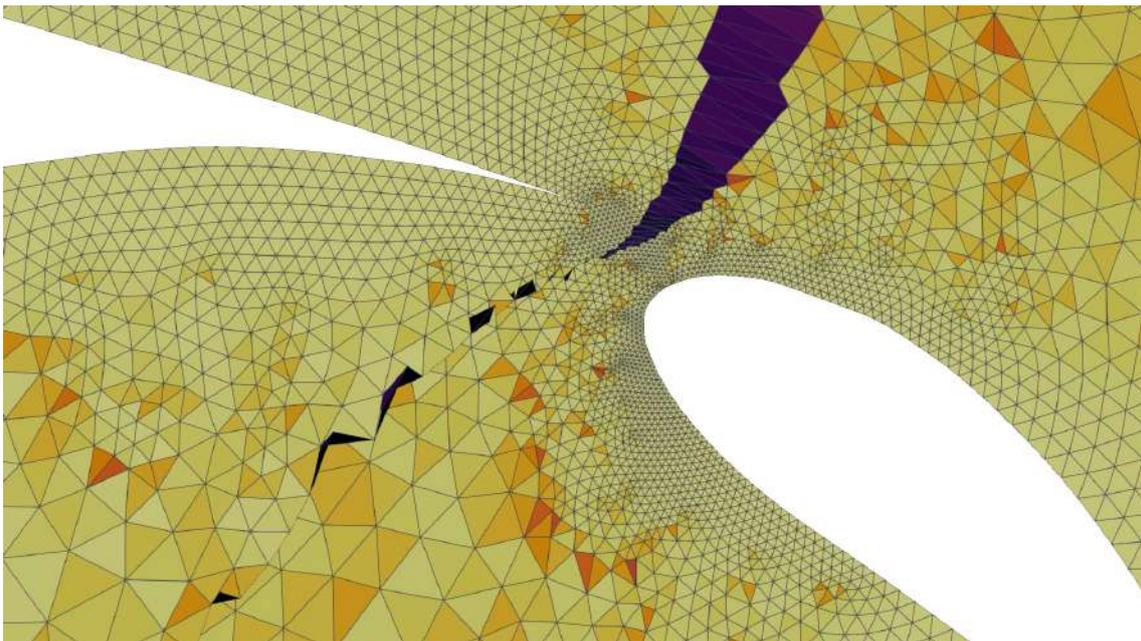
Figure 7.7: Application of the original RBF mesh deformation method around a four-element airfoil. Slats and flaps of the airfoil move upwards and downwards. Wendland C^2 with $R_s = 5$ was chosen as the activation function of the RBF network.

7.2.4 Wendland C^0 ($R_s = 5$) with Exponential weighting function

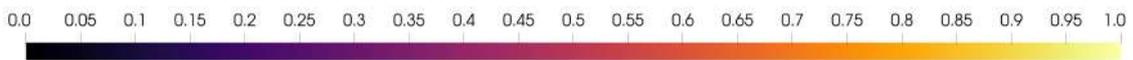
At this stage the proposed hybrid method is applied. Wendland C^0 function with a support radius $R_s = 5$ is used as the activation function of the RBF network while exponential is the selected weighting function. As it is depicted in Figure 7.8 inverted elements appear between moving and stationary bodies. This is the case for all combinations of activation and weighting functions that were tested.



(a) Between moving and stationary elements inappropriate elements appear



(b) Region between the main element and the first flap



(c) Size-shape metric

Figure 7.8: Application of the hybrid RBF mesh deformation method around a four-element airfoil. Slats and flaps of the airfoil move downwards. Wendland C^0 with $R_s = 5$ was chosen as the activation function of the RBF network and exponential function was the selected weighting function. The hybrid method fails to deform the mesh appropriately.

Chapter 8

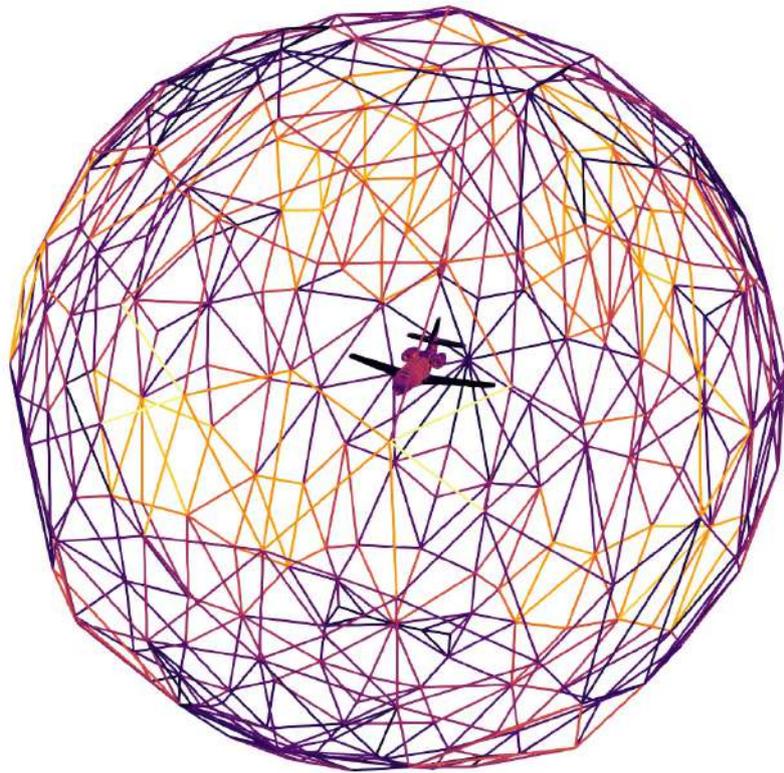
Application to the mesh around an aircraft

8.1 Problem description

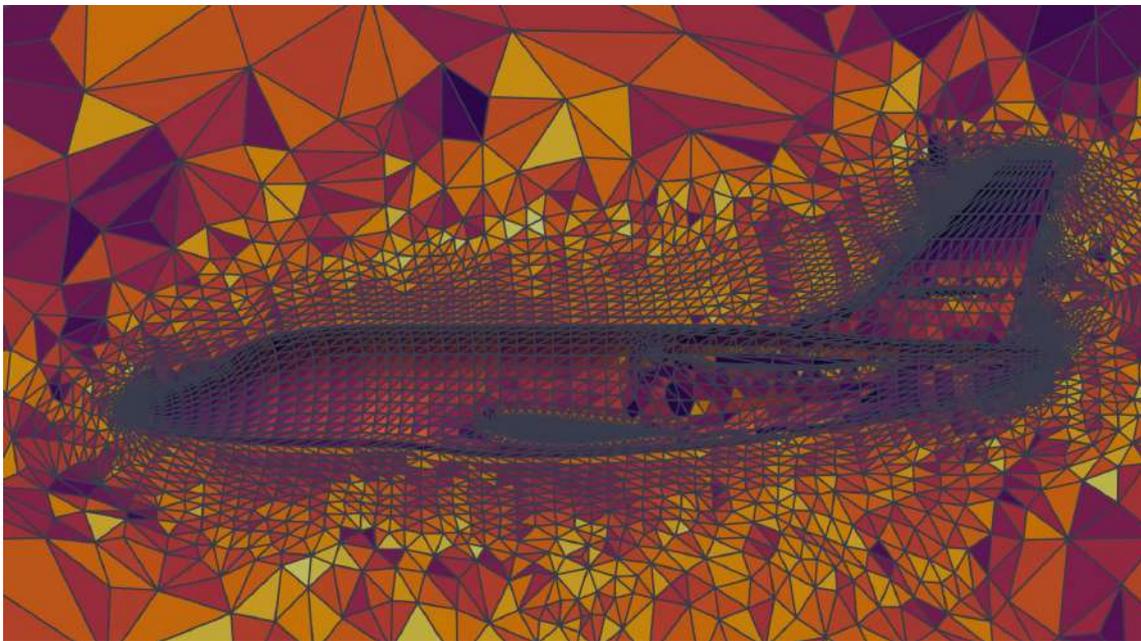
In this chapter, the RBF method and the proposed hybrid method are applied to deform the mesh around an aircraft within a stationary bounding sphere. Between the aircraft and the outer sphere, a mesh with tetrahedral elements is generated using the advanced-front method [36]. The mesh is provided in two files (i.e., LTT format: one file with nodes coordinates and one file with nodes connectivity), and it is then transformed in VTK file format in order to be compatible with ParaView [37]. For this purpose, software that converts both 2D and 3D LTT unstructured meshes into VTK format was developed. Additionally, every element of the mesh is colored according to its shape quality (see Appendix A) through a color-quality mapping. Figure 8.1 depicts the original mesh as it is visualized in ParaView.

An increasing pitch angle is applied to the aircraft, and subsequently, the internal mesh is deformed, conforming to the updated boundary nodes locations. Original RBF method using Wendland C^0 and Wendland C^2 as activation functions (see Table 2.2).

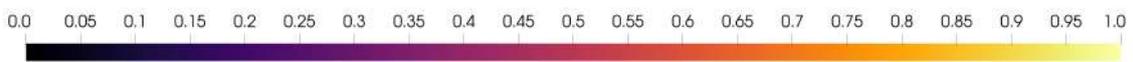
The unstructured mesh around the aircraft consists of 87965 nodes, of which the 5836 are boundary, and form 511888 tetrahedra. The whole set of the 5836 boundary nodes constitute the set of the centers that train the RBF network.



(a) *Bounding sphere and the surface mesh on the aircraft*

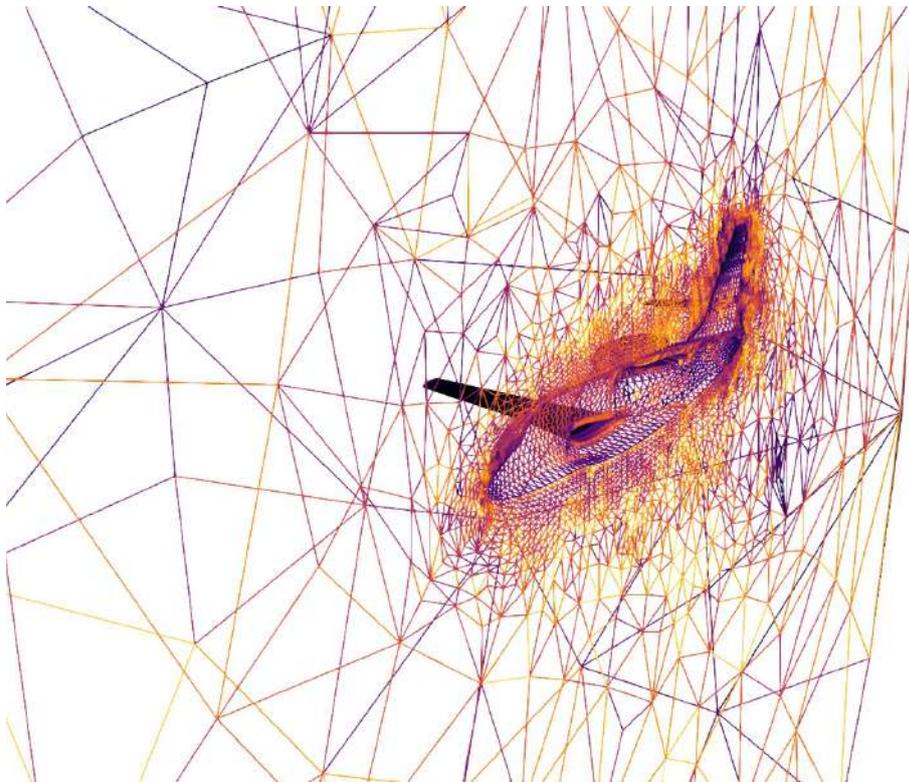
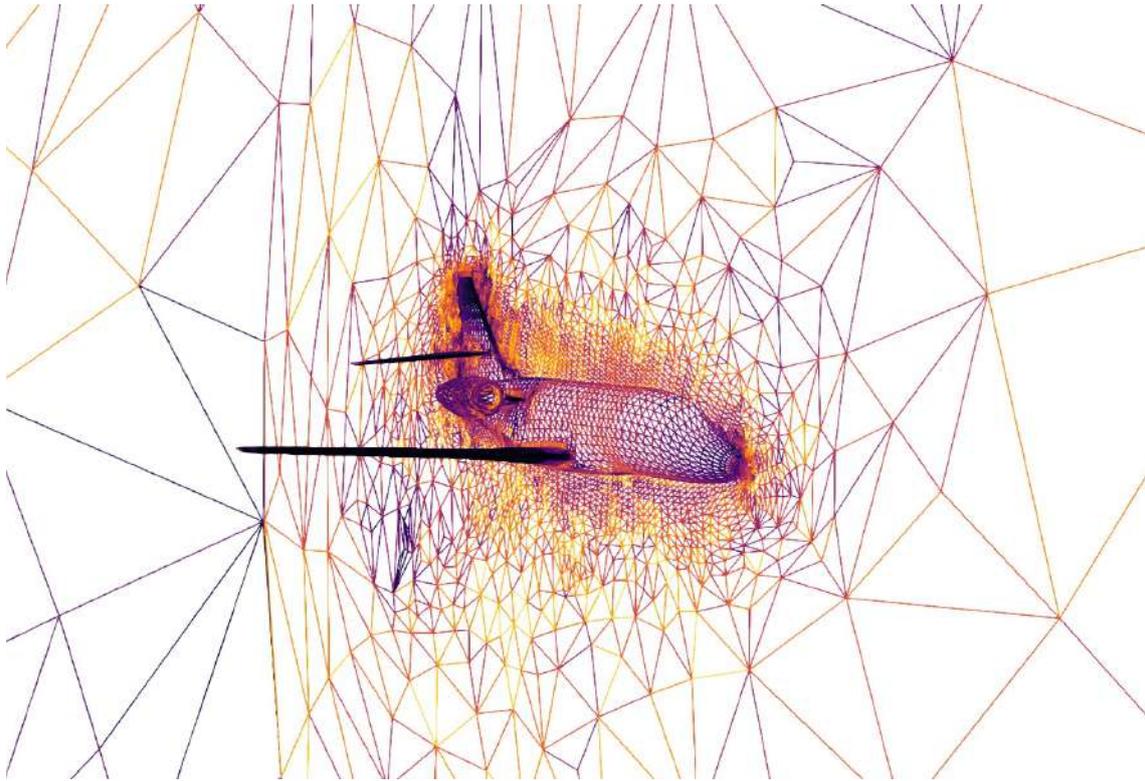


(b) *Sectional view of the aircraft and the internal mesh around it*



(c) *Size-shape metric*

Figure 8.1: *The original mesh with tetrahedral elements around an aircraft*



(a) *Size-shape metric*

Figure 8.2: *Sectional view of the aircraft and its mesh from a different viewpoint* 85

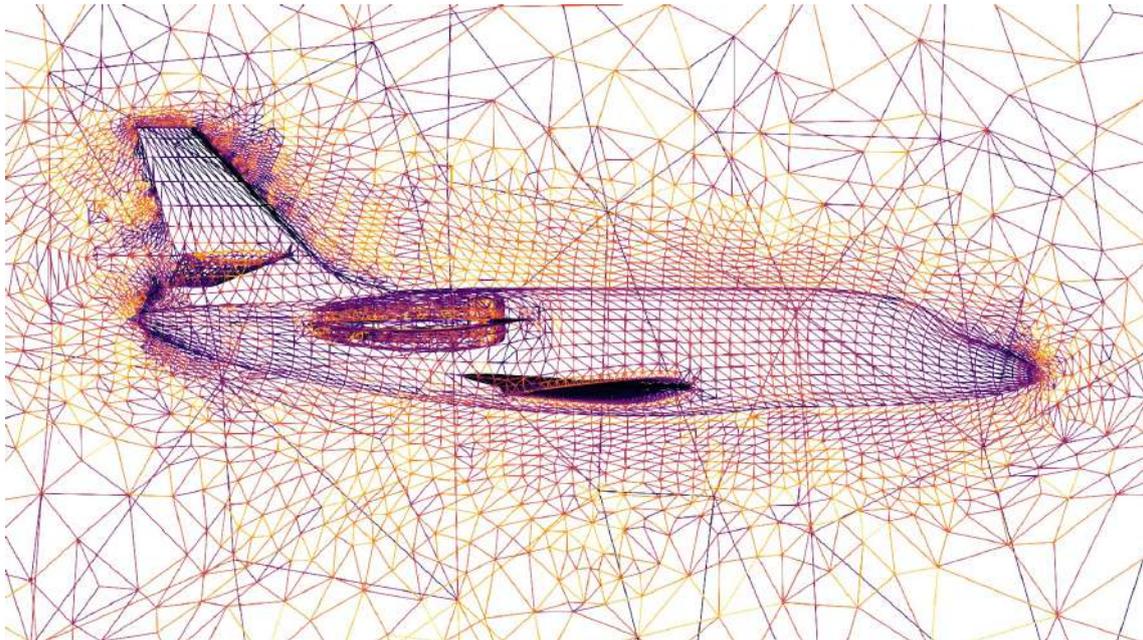
8.2 Results

8.2.1 Wendland C^0

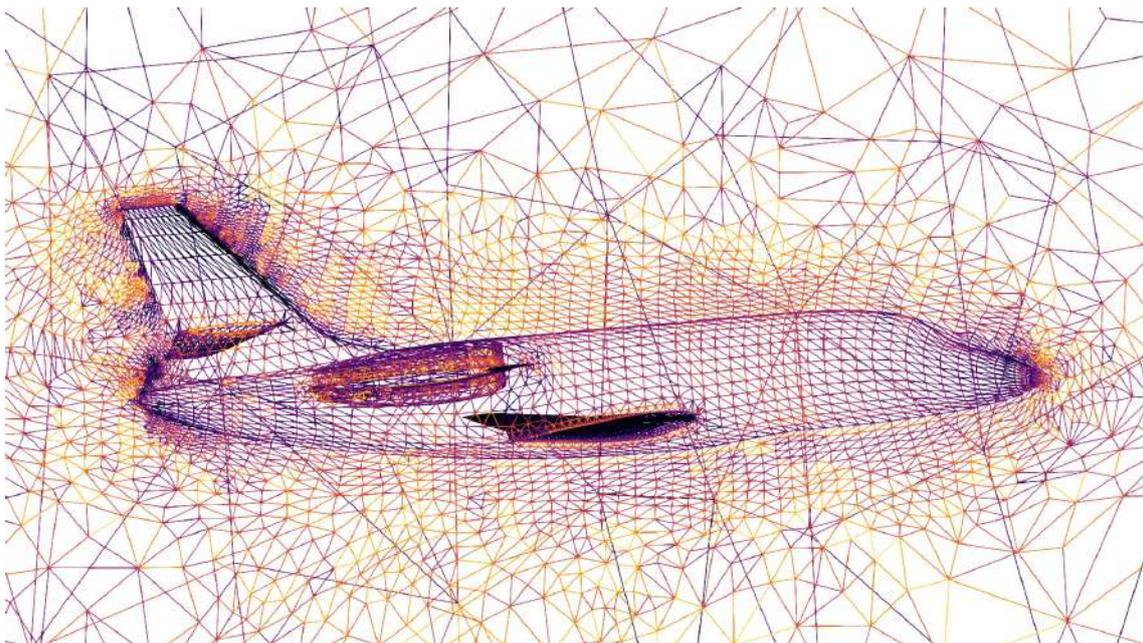
In this section the locally supported Wendland C^0 function (see Table 2.2) was chosen as activation function of the RBF network. The aircraft rotates along pitch axis for 5 degrees. The activation function is deployed with a radius of support equal to 40. For pitch angles greater than 5 degrees inverted elements appear on the deformed mesh. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for pitch angle equal to 5 degrees are presented in Table 8.1.

$\phi(r)$	5
$\overline{f_{ss}}$	0.368
$\sigma_{f_{ss}}$	0.216
$f_{ss,\min}$	0.002

Table 8.1: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the aircraft. Wendland C^0 with support radius $R_s = 40$ is the selected RBF kernel.



(a) *Initial mesh*



(b) *Deformed mesh for rotation of 5 degrees along y-axis*



(c) *Size-shape metric*

Figure 8.3: Application of the original RBF mesh deformation method around an aircraft. The aircraft rotates 5 degrees around the pitch axis. Wendland C^0 function with $R_s = 40$ was chosen as the activation function with compact support of the RBF network

8.2.2 Wendland C^2

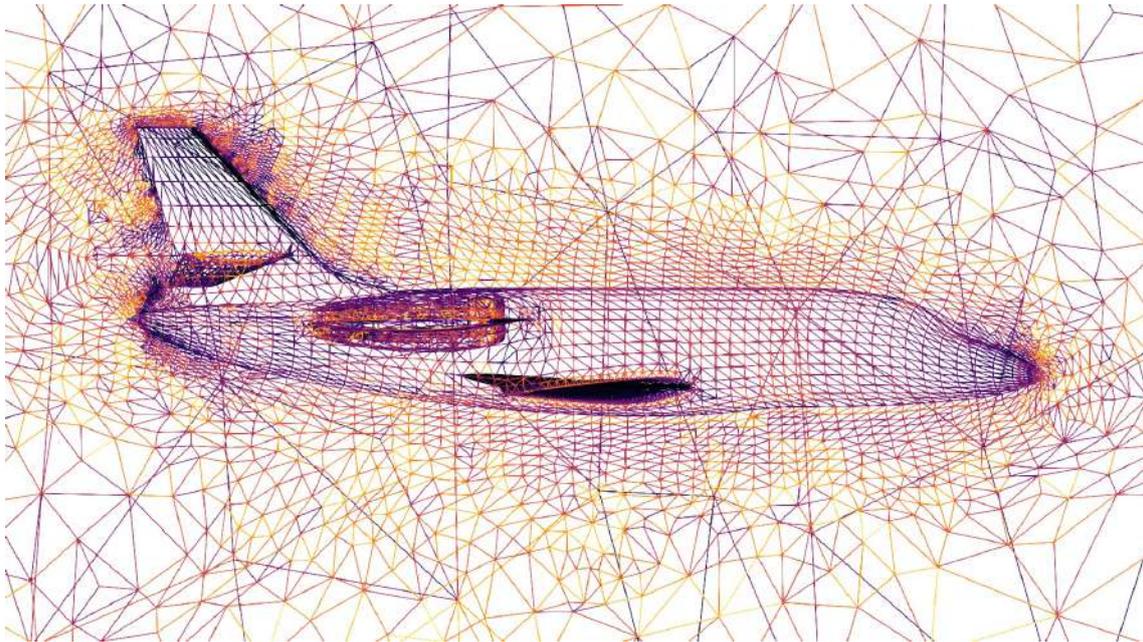
Hereupon the locally supported Wendland C^2 function (see Table 2.2) was chosen as activation function of the RBF network. The aircraft is repeatedly rotated along pitch axis until inverted elements appear. The activation function is deployed with two radius of support $R_s = 21$ and $R_s = 40$.

Support radius $R_s = 21$

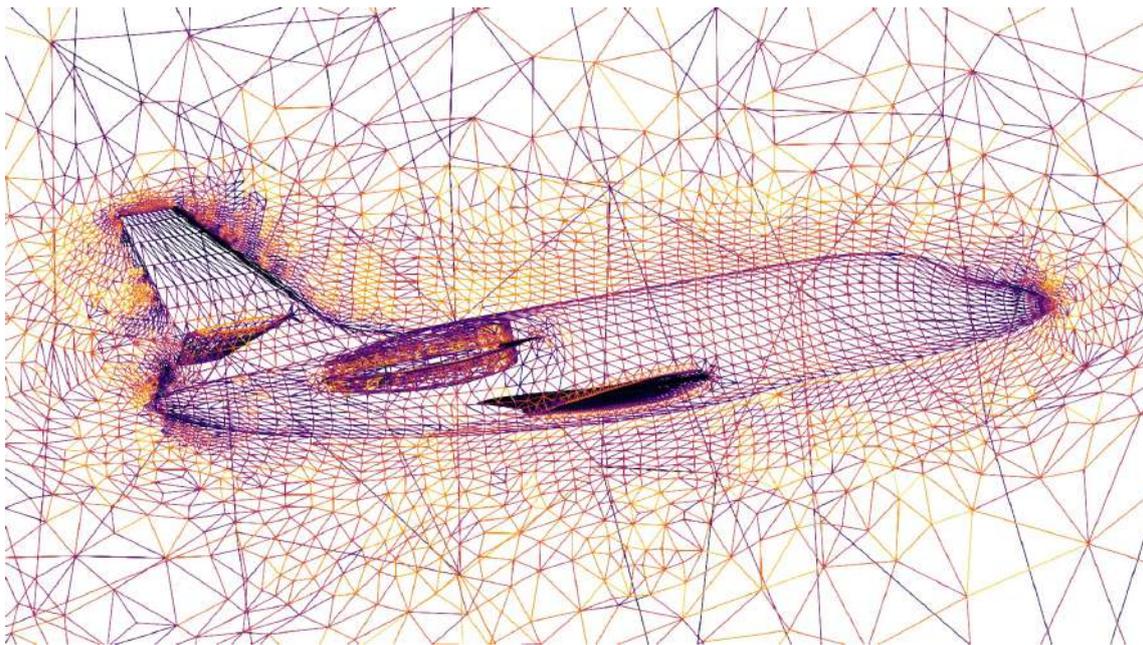
At first radius of support is set equal to 21 in order to enclose at least the whole boundary of the aircraft. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for pitch angles equal to 5, 8, 10 and 12 degrees are presented in Table 8.2.

$\phi(r)$	5	8	10	12
$\overline{f_{ss}}$	0.367	0.365	0.363	0.362
$\sigma_{f_{ss}}$	0.215	0.213	0.212	0.211
$f_{ss,\min}$	0.002	0.002	0.002	0.002

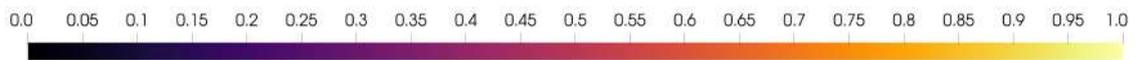
Table 8.2: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the aircraft. Wendland C^2 with support radius $R_s = 21$ is the selected RBF kernel.



(a) *Initial mesh*



(b) *Deformed mesh for rotation of 10 degrees along pitch axis*



(c) *Size-shape metric*

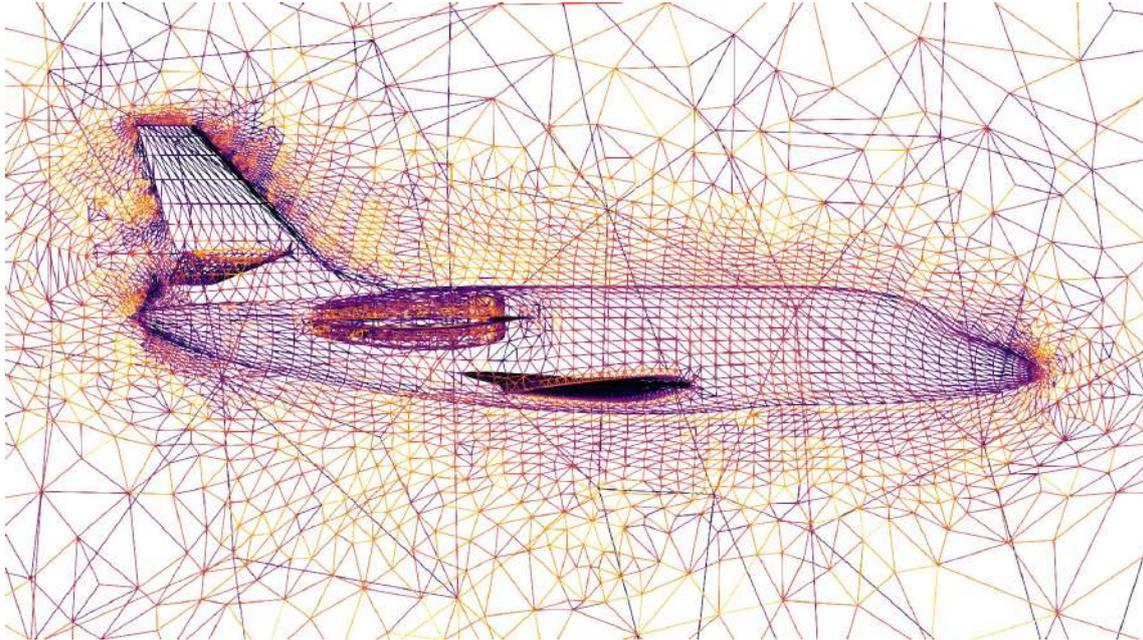
Figure 8.4: Application of the original RBF mesh deformation method around an aircraft. The aircraft rotates 10 degrees around the pitch axis. Wendland C^2 function with $R_s = 21$ was chosen as the activation function with compact support of the RBF network

Support radius $R_s = 40$

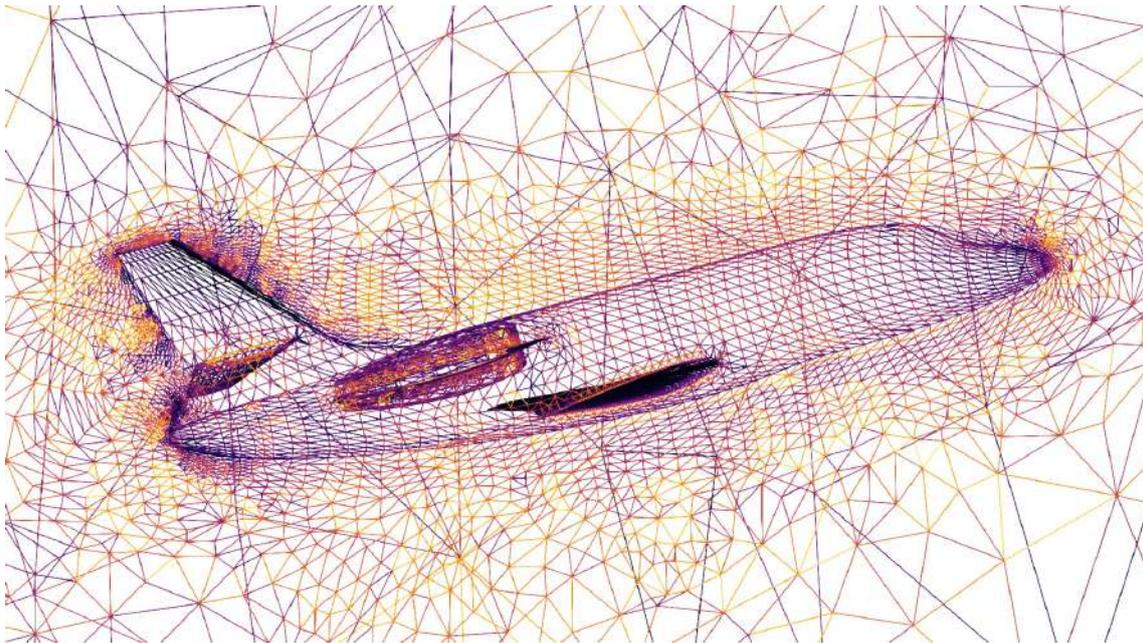
Here radius of support is increased to 40 in order to achieve a greater angle of rotation. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for pitch angles equal to 5, 10, 12 and 15 degrees are presented in Table 8.3. For angles greater than 15 degrees inverted elements appear.

$\phi(\mathbf{r})$	5	10	12	15
$\overline{f_{ss}}$	0.369	0.367	0.367	0.367
$\sigma_{f_{ss}}$	0.216	0.216	0.215	0.215
$f_{ss,\min}$	0.002	0.002	0.002	0.002

Table 8.3: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the aircraft. Wendland C^2 with support radius $R_s = 21$ is the selected RBF kernel.



(a) *Initial mesh*



(b) *Deformed mesh for rotation of 15 degrees along pitch axis*



(c) *Size-shape metric*

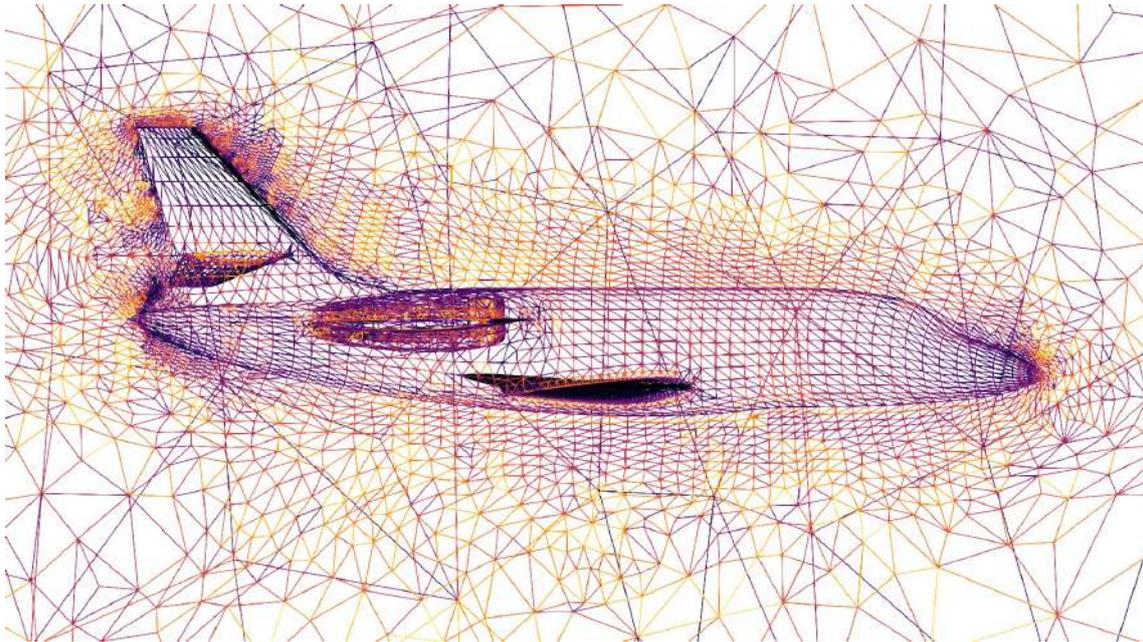
Figure 8.5: Application of the original RBF mesh deformation method around an aircraft. The aircraft rotates 15 degrees around the pitch axis. Wendland C^2 function with $R_s = 40$ was chosen as the activation function with compact support of the RBF network

8.2.3 Wendland C^2 ($R_s = 40$) with Gaussian as weighting function

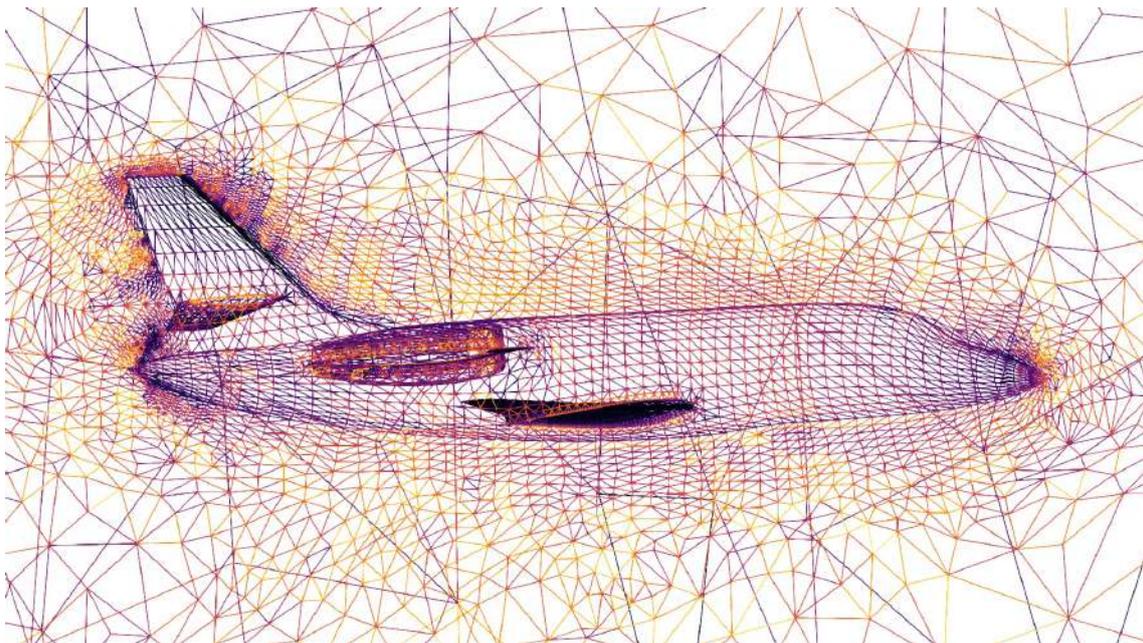
At this stage the proposed hybrid method is applied. Wendland C^0 function with a support radius $R_s = 40$ is used as the activation function of the RBF network while exponential is the selected weighting function. The average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric that is introduced in Appendix A for pitch angle equal to 3 degrees are presented in Table 8.4. For angles greater than 4 degrees inverted elements appear.

$\phi(r)$	5
$\overline{f_{ss}}$	0.369
$\sigma_{f_{ss}}$	0.217
$f_{ss,\min}$	$3.99 \cdot 10^{-5}$

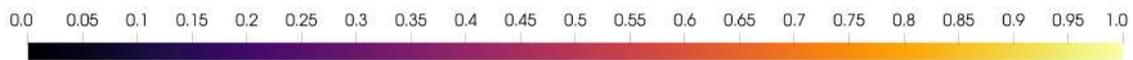
Table 8.4: Average value, standard deviation and minimum value of the mesh quality metric for each angle of rotation of the aircraft. Wendland C^2 with support radius $R_s = 40$ is the selected RBF kernel and exponential function is the weighting function.



(a) *Initial mesh*



(b) *Deformed mesh for rotation of 3 degrees along pitch axis*



(c) *Size-shape metric*

Figure 8.6: Application of the original RBF mesh deformation method around an aircraft. The aircraft rotates 3 degrees around the pitch axis. Wendland C^2 function with $R_s = 40$ was chosen as the activation function with compact support of the RBF network while exponential was the selected weighting function.

Chapter 9

Aggregate results

This chapter compares the two methods together with the various activation and weighting functions that were deployed previously.

9.1 Rotated sycamore Leaf

In this section the two methods together with the various activation functions are compared. In Table 9.1 the average value ($\overline{f_{ss}}$), the standard deviation ($\sigma_{f_{ss}}$) and the minimum value ($f_{ss,\min}$) of the mesh quality metric, for an angle of rotation equal to 30 degrees, are presented.

Function	$\overline{f_{ss}}$	$\sigma_{f_{ss}}$	$f_{ss,\min}$
Inverse quadratic	0.696	0.258	0
Inverse multiquadric	0.762	0.187	0.080
Wendland C^0 ($R_s = 11$)	0.811	0.113	0.276
Wendland C^2 ($R_s = 11$)	0.843	0.120	0.177
Wendland C^2 ($R_s = 11$) (hybrid)	0.857	0.140	0.067

Table 9.1: Average value, standard deviation and minimum value of the mesh quality metric for angle of rotation of the sycamore leaf equal to 30 degrees.

It is seen that compactly supported RBFs achieve better results. Regarding the hybrid method, the deformed mesh is slightly of a better quality. It should be noted that the main asset of that approach is that near the solid boundary the deformed mesh is almost of similar quality of initial mesh. This is quite important as at that regions great gradients of pressure and velocity occur and thus the mesh should be as of higher quality as it gets in order the flow to be simulated properly.

9.2 Rotated four-element airfoil

Hereupon, the same procedure is followed for the mesh around a rotated four-element airfoil. This time the angle of rotation is set to be equal to 60 degrees as the present mesh can conform to bigger angles of rotation.

Function	$\overline{f_{ss}}$	$\sigma_{f_{ss}}$	$f_{ss,min}$
Exponential	0.822	0.150	0.045
Wendland C^0 ($R_s = 5$)	0.807	0.164	0.036
Wendland C^0 ($R_s = 19$)	0.825	0.147	0.046
Wendland C^2 ($R_s = 10$)	0.966	0.063	0.097
Wendland C^2 ($R_s = 2$) (hybrid)	0.970	0.055	0.099

Table 9.2: Average value, standard deviation and minimum value of the mesh quality metric for angle of rotation of the four-element airfoil equal to 60 degrees.

Regarding the typical RBF method, Wendland C^2 presents the most robust behavior. Even for angles of rotation equal to 90 degrees the mesh does not presents inverted elements. The hybrid RBF method achieves similar results when Wendland C^0 is used as weighting function. Although the deformed mesh does not present an improvement regarding the average size-shape metric compared to typical RBF method, the accomplishment of the hybrid method is the fact that can produce deformed meshes of same or better quality but using a relative smaller support radius for the RBF kernel. The latter leads to sparse interpolation matrices which can be solved more computationally efficiently than dense ones.

9.3 Aircraft

Lastly the results that derive from the application to the mesh around a three-dimensional aircraft are presented. At this case the mesh contains a quite large number of elements (511888 tetrahedra). Therefore it does not make sense to compare its quality metrics as they arise after the application of the various functions and methods. It is more meaningful to see the maximum angle of rotation that a method can achieve without inverted elements appear as it is shown in Table 9.3

Function	ϕ_{crit}
Wendland C^0 ($R_s = 40$)	5°
Wendland C^2 ($R_s = 21$)	12°
Wendland C^2 ($R_s = 40$)	16°
Wendland C^2 ($R_s = 40$) (hybrid)	4°

Table 9.3: *Critical angles of rotation*

The standard RBF method with Wendland C^2 as activation function is clearly the most robust. Especially when it is applied with a sufficiently large radius of influence, it can support angles of rotation equal to 16 degrees.

Chapter 10

Summary - Conclusion

The present diploma thesis focuses on developing and programming a mesh deformation algorithm that uses an RBF network and quaternions defined on boundary nodes. The need for adapting an existing mesh to displaced boundaries arises in many applications (e.g., shape optimization, aeroelasticity problems, flow simulations in the presence of moving bodies). For industrial-scale applications, remeshing is highly costly, and it might also hinder the continuation of new simulations from available numerical solutions on the unstructured mesh. Therefore, an automated algorithm that morphs the existing mesh is used instead.

Quaternions are four-element vectors that are ideal for modeling rotations in three-dimensional space. RBF networks use radial basis functions as activation functions, and they are extensively employed in mesh morphing applications. The proposed hybrid approach couples the known RBF method with quaternions in order to produce a higher quality mesh near the boundary, where greater gradients of pressure and velocity occur.

The training of the RBF network requires solving two (for 2D cases) or three (for 3D cases) linear systems. These linear systems have the same coefficient matrix, which is also symmetric, and thus Cholesky decomposition is applied in order to solve them. The Quaternions are defined on the source nodes (i.e., boundary nodes) by comparing the orientation of the boundary normals between undeformed and deformed mesh. Subsequently, the boundary displacement is propagated into the internal mesh nodes as a weighted sum of the displacement of the closest boundary node and the displacement that emerges typically from the RBF method. The weights on the sum mentioned above are determined from a weighting function. The weighting function is deployed in such a way that quaternions have strong influence near the boundary which erodes as the distance increases. Therefore, the primary condition is the output of the function to decrease smoothly as distance increases.

Both methods were tested using various combinations of activation and weighting functions. The deformed meshes were presented together with their quality metrics. In order to be able to evaluate the various meshes, each element is colored through a color-quality mapping.

Appendix A

Measuring mesh quality

A.1 Impact of mesh quality

Arguably, mesh quality can have a significant influence on the accuracy and efficiency of numerical simulations. A high-quality mesh can lead to faster convergence of the solver and accurate simulation results. While increasing the fineness level will typically yield a higher quality mesh, the computational cost will also significantly increase. Therefore, it is safe to say that the aim is to achieve an optimal balance between the computational cost and the level of fineness. Correspondingly, a low-quality mesh will not only contribute to inaccurate simulation results but might even cause the solver to produce errors due to instability. Poorly shaped elements can give rise to ill-conditioned matrices, which tend to slow or even cause divergence of iterative solvers.

It follows that the use of mesh quality metrics is an essential part of mesh generation. A minimal requirement for mesh quality is that all the elements be non-inverted. Beyond that, it is desirable that the elements have a good shape and size to reduce truncation error. Ideally, the definition of good shape and good size should be tied to solution error, meaning that the total mesh quality is good enough if the error is below some acceptable level. In practice, the solution comes after the generation of the initial mesh, and usually, it is not available to the person who performs the mesh generation. The absence of comprehensive knowledge about the solution leads to a priori quality metrics that perform quality control to the produces meshes. This is done by considering that the geometrical properties of the mesh somehow correspond to the downstream simulation accuracy. Therefore, numerous geometrically-based quality metrics have been developed. These quality metrics take into account several aspects related to the elements of the mesh, such as size, orientation, shape and skew. The Jacobian matrix plays a central role in the above metrics as it contains

information on all the above aspects.

A.2 Mesh quality metrics

In the present diploma thesis, mesh deformation methods are studied in order to avoid the computationally expensive re-meshing procedure. So, it is crucial to be able to compare the quality of different meshes after mesh movement. For this purpose, the mesh quality metrics proposed by Knupp [39] are used. It is assumed that the initial mesh provided from PCOpt/NTUA was generated optimally, and hence the element shapes should be deformed as little as possible. This means that both the shape and the size should be preserved. A mixed relative size and shape metric is introduced in order to measure both of the above quantities.

A.2.1 Triangle relative size metric

The relative size metric detects elements that are unusually large or unusually small compared to a reference element. As described above, the reference element is selected to be the corresponding element of the initial mesh; hence it is clear that the relative size metric measures the change in element size.

Let τ be the ratio of the area of the current triangle to the area of the corresponding triangle of the initial mesh. The relative size metric [39] is then given by

$$f_{\text{size}} = \min\left(\tau, \frac{1}{\tau}\right) \quad (\text{A.1})$$

This metric is dimensionless and has the minor drawback that is not everywhere differentiable with respect to node position. A differentiable workaround have been also proposed in [39] and is presented bellow for the sake of completeness.

$$f_{\text{size}} = \frac{2\tau}{1 + \tau^2} \quad (\text{A.2})$$

From Equation A.1 some essential properties arise. Value of f_{size} equals one if and only if the deformed triangle has the same area as the initial triangle and $f_{\text{size}} = 0$ if and only if the deformed triangle is degenerate. Additionally, it should be noted that the value of f_{size} only depends on the change of the area of the current triangle. It is symmetric with respect to the sign of this change meaning that a diminution is emphasized as much as an enlargement. To illustrate this point, a deformed triangle whose area has been doubled will produce the same value of f_{size} as a deformed triangle whose area is half the area of the initial triangle. However, a change in the shape of the triangle does not affect in any way the relative size metric meaning that badly deformed elements can preserve their size and hence achieve a value of relative

size metric near one. Because of this property, relative size metric is probably not that convenient when it is used as a standalone. As a more handy approach the combined size-shape metric is chosen.

A.2.2 Triangle shape metric

Shape quality metric is deployed in order to detect distortions in the shape of a triangular cell without take under consideration its size. The most common approach is to assume that the mesh will be used in simulations which contain isotropic physics, hence the ideal element for the shape metric is the equilateral triangle. A shape quality metric with this property is presented bellow

$$f_{\text{shape}} = \frac{3r \sin \theta}{1 - r \cos \theta + r^2} \quad (\text{A.3})$$

where $r = \sqrt{\lambda_{22}/\lambda_{11}}$ is the ratio of consecutive lengths and θ is the angle between the two sides joined at the specific node. Therefore, shape metric is a combination of both element angles and length ratios.

From Equation A.3 some properties specific to the triangle shape metric arise. Value of f_{shape} equals one if and only if the physical triangle is equilateral and, similar to f_{size} , $f_{\text{shape}} = 0$ if and only if the deformed triangle is degenerate.



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής
& Βελτιστοποίησης

Προγραμματισμός Μεθόδου Παραμόρφωσης Πλέγματος με Συναρτήσεις Ακτινικής Βάσης και Τετραδόνια

Εκτενής Περίληψη Διπλωματικής Εργασίας

Βασίλειος Γεωργόπουλος

Επιβλέπων
Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2021

Παραμόρφωση Πλέγματος με Δίκτυα RBF

Τα δίκτυα συναρτήσεων ακτινικής βάσης (RBF) χρησιμοποιούνται ευρέως σε εφαρμογές όπου είναι απαραίτητη η προσαρμογή του υφιστάμενου υπολογιστικού πλέγματος. Το δίκτυο αυτό εκπαιδεύεται βάσει των γνωστών μετατοπίσεων των οριακών κόμβων του πλέγματος και κατόπιν εφαρμόζεται σε όλους τους εσωτερικούς κόμβους του πλέγματος των οποίων η μετατόπιση είναι άγνωστη. Η εκπαίδευση του δικτύου γίνεται με τον υπολογισμό των βαρών $\mathbf{w}_1, \dots, \mathbf{w}_N$, έτσι ώστε αυτό να παρεμβάλει ακριβώς τις τιμές των μετατοπίσεων των οριακών κόμβων. Αυτό απαιτεί την επίλυση δύο (αν το πρόβλημα είναι διδιάστατο) ή τριών (αν το πρόβλημα είναι τριδιάστατο) γραμμικών $N \times N$ συστημάτων,

$$\underbrace{\begin{bmatrix} \phi_1(\mathbf{x}_{s_1}) & \dots & \phi_N(\mathbf{x}_{s_1}) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_{s_N}) & \dots & \phi_N(\mathbf{x}_{s_N}) \end{bmatrix}}_{\Phi_s} \underbrace{\begin{bmatrix} \mathbf{w}_1^\top \\ \vdots \\ \mathbf{w}_N^\top \end{bmatrix}}_W = \underbrace{\begin{bmatrix} \delta_1^\top \\ \vdots \\ \delta_N^\top \end{bmatrix}}_{\Delta_s} \quad (4)$$

Το μητρώο των σταθερών συντελεστών Φ_s είναι συμμετρικό, θετικά ορισμένο και στη διαγώνιο του έχει παντού τιμή ίση με τη μονάδα. Επιπλέον, είναι το ίδιο για κάθε κατεύθυνση πράγμα που σημαίνει ότι η υπολογιστικά κοστοβόρα διαδικασία εύρεσης του αντίστροφου του (λ.χ. με τη μέθοδο του Cholesky) χρειάζεται να γίνει μόνο μια φορά. Έτσι τα βάρη υπολογίζονται

$$\mathbf{w}_x = \Phi^{-1} \Delta_x, \quad \mathbf{w}_y = \Phi^{-1} \Delta_y, \quad \mathbf{w}_z = \Phi^{-1} \Delta_z \quad (5)$$

Ακολούθως, αφού έχουν υπολογιστεί τα βάρη του δικτύου RBF, αυτό εφαρμόζεται σε κάθε εσωτερικό κόμβο και έτσι προκύπτουν οι μετατοπίσεις στο σύνολο του πλέγματος.

$$\delta_x(\mathbf{x}) = \sum_{n=1}^N w_{x_n} \phi(\|\mathbf{x} - \mathbf{x}_{s_n}\|) \quad (6)$$

$$\delta_y(\mathbf{x}) = \sum_{n=1}^N w_{y_n} \phi(\|\mathbf{x} - \mathbf{x}_{s_n}\|) \quad (7)$$

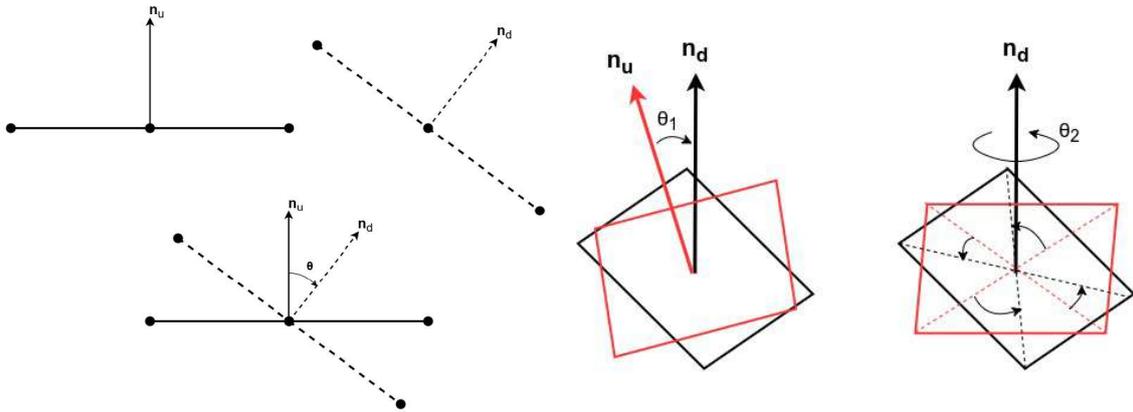
$$\delta_z(\mathbf{x}) = \sum_{n=1}^N w_{z_n} \phi(\|\mathbf{x} - \mathbf{x}_{s_n}\|) \quad (8)$$

Υβριδική μέθοδος με χρήση τετραδονίων

Στη διπλωματική αυτή εργασία προτείνεται μια υβριδική μέθοδος η οποία συνδυάζει την τυπική μέθοδο RBF με τετραδόνια τα οποία ορίζονται πάνω στους οριακούς κόμ-

βους του πλέγματος. Τα τετραδόνια παρουσιάστηκαν για πρώτη φορά από τον Ιρλανδό μαθηματικό W.R. Hamilton το 1843 και ουσιαστικά αποτελούν μια επέκταση των μιγαδικών αριθμών στον τριδιάστατο χώρο. Είναι διανύσματα τεσσάρων στοιχείων ($Q = q_0 + q_1i + q_2j + q_3k$, ή $Q = [q_0, \mathbf{q}]$) τα οποία βρίσκουν ευρεία εφαρμογή στη μοντελοποίηση περιστροφών, ειδικά στις τρεις διαστάσεις. Όπως είναι γνωστό κάθε στροφή στο χώρο μπορεί να περιγραφεί πλήρως από μια γωνία θ και από έναν άξονα περιστροφής \mathbf{u} . Η πληροφορία αυτή αποθηκεύεται σε ένα τετραδόνιο ως: $Q_{rot} = [\cos \frac{\theta}{2}, \mathbf{u} \sin \frac{\theta}{2}]$. Η περιστροφή ενός σημείου $P = [0, \mathbf{p}]$ μπορεί πλέον να εκφραστεί με χρήση των τετραδονίων ως: $P_{rot} = Q_{rot} P Q_{rot}^{-1}$.

Η κατασκευή των τετραδονίων πάνω στους οριακούς κόμβους γίνεται συγκρίνοντας των προσανατολισμό των οριακών ακμών (στις δύο διαστάσεις) ή των κελιών (στις τρεις διαστάσεις) πριν και μετά την παραμόρφωση. Κατασκευάζεται το εκάστοτε τετραδόνιο το οποίο όταν εφαρμοστεί στο κόμβο του μη-παραμορφωμένου πλέγματος θα τον περιστρέψει κατάλληλα έτσι ώστε οι δύο ακμές να έχουν τον ίδιο προσανατολισμό ή τα δύο κελιά να βρίσκονται πάνω στο ίδιο επίπεδο. Στις τριδιάστατες περιπτώσεις είναι απαραίτητη άλλη μια στροφή έτσι ώστε η γωνία μεταξύ των διανυσμάτων που ενώνουν τον εκάστοτε κόμβο με τους γειτονικούς πριν και μετά την παραμόρφωση να είναι ελάχιστη.



Σχήμα 1: Διαδικασία κατασκευής οριακών τετραδονίων για τη διδιάστατη (αριστερά) και την τριδιάστατη (δεξιά) περίπτωση.

Σε επόμενο στάδιο υπολογίζεται η απόσταση μεταξύ κάθε εσωτερικού κόμβου του πλέγματος και του πλησιέστερου οριακού κόμβου. Ο εκάστοτε εσωτερικός κόμβος μετατοπίζεται επηρεαζόμενος από τη μετατόπιση (η οποία έχει εκφραστεί με τετραδόνια) του πλησιέστερου οριακού κόμβου. Η τελική μετατόπιση του κάθε εσωτερικού κόμβου προκύπτει από το σταθμισμένο άθροισμα της μετατόπισης όπως προκύπτει από το δίκτυο RBF και της μετατόπισης του πλησιέστερου οριακού κόμβου.

$$\delta_{internal} = \gamma \delta_{RBF} + (1 - \gamma) \delta_{closest\ node} \quad (9)$$

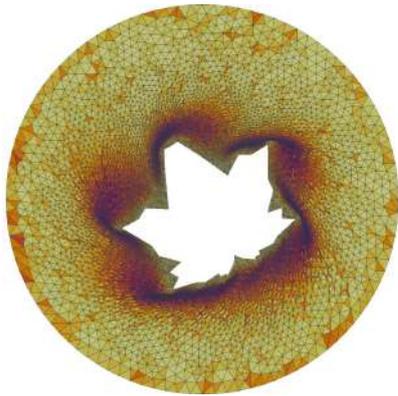
Η συνάρτηση βάρους $\gamma = \gamma(d)$ επιλέγεται κατάλληλα έτσι ώστε κοντά στο όριο να κυριαρχεί η μετατόπιση βάσει των τετραδονίων ενώ μακριά από το όριο να κυριαρχεί η μετατόπιση όπως αυτή προκύπτει από το δίκτυο RBF.

Εφαρμογή σε πλέγμα γύρω από φύλλο πλατάνου

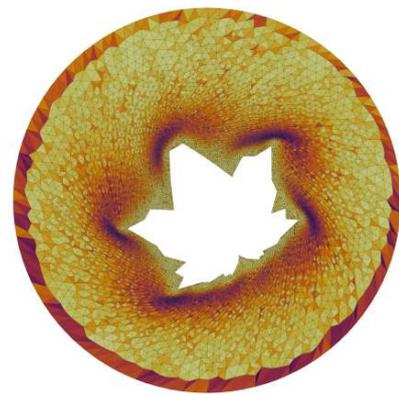
Στην ενότητα αυτή, η τυπική μέθοδος RBF καθώς και η προτεινόμενη υβριδική μέθοδος εφαρμόζονται στο πλέγμα γύρω από ένα φύλλο πλατάνου. Πραγματοποιούνται διαφορετικού μέτρου στροφές του φύλλου γύρω από την αρχή των αξόνων και ακολούθως το αρχικό πλέγμα μετατοπίζεται κατάλληλα σύμφωνα με τη μετατόπιση του ορίου. Στον Πίνακα 1 παρουσιάζονται τα αποτελέσματα της μετρικής ποιότητας του πλέγματος σε στροφή 30° για διάφορες συναρτήσεις ενεργοποίησης.

Συνάρτηση	$\overline{f_{ss}}$	$\sigma_{f_{ss}}$	$f_{ss,\min}$
Inverse quadratic	0.696	0.258	0
Inverse multiquadric	0.762	0.187	0.080
Wendland C^0 ($R_s = 11$)	0.811	0.113	0.276
Wendland C^2 ($R_s = 11$)	0.843	0.120	0.177
Wendland C^2 ($R_s = 11$) (hybrid)	0.857	0.140	0.067

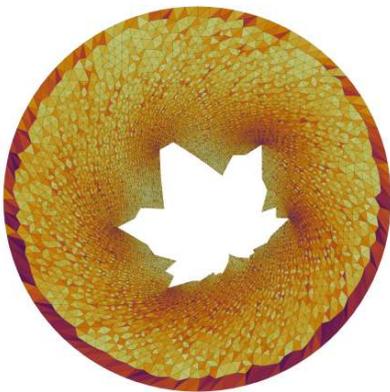
Πίνακας 1: Μέση τιμή, τυπική απόκλιση και ελάχιστη τιμή της μετρικής ποιότητας πλέγματος για διάφορες συναρτήσεις ενεργοποίησης. Η υβριδική μέθοδος χρησιμοποιεί την συνάρτηση $Wendland C^0$ ως συνάρτηση βάρους.



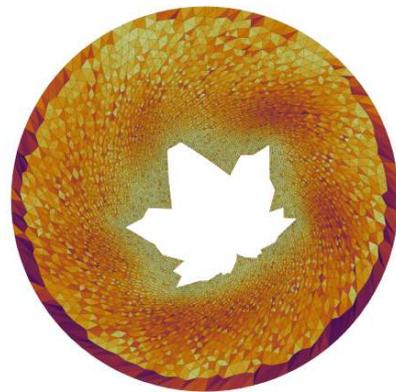
(α') *Inverse quadratic*



(β') *Inverse multiquadric*

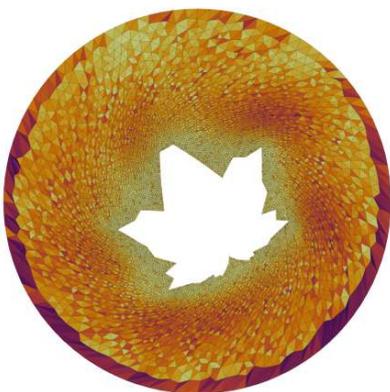


(γ') *Wendland C^0*

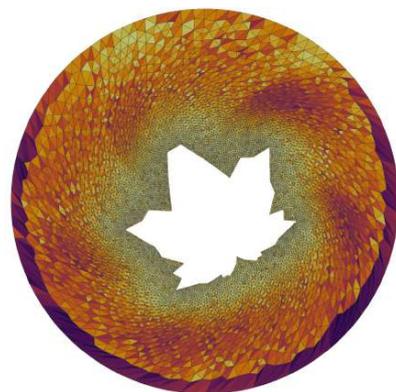


(δ') *Wendland C^2*

Σχήμα 2: Προκύπτοντα πλέγματα για στροφή 30°



(α') *Wendland C^2 original*



(β') *Wendland C^2 hybrid*

Σχήμα 3: Προκύπτοντα πλέγματα για στροφή 30° . Σύγκριση μεταξύ τυπικής και υβριδικής μεθόδου.

Στο Σχήμα 2 παρατηρείται ότι οι συναρτήσεις τοπικής επιρροής Wendland C^0 και Wendland C^2 οδηγούν σε σαφώς ποιοτικότερα πλέγματα συγκριτικά με τις συναρτήσεις ολικής επιρροής Inverse quadratic και Inverse multiquadric. Ειδικότερα στις συναρτήσεις ολικής επιρροής το πλέγμα χειροτερεύει εμφανώς κοντά στις ακμές του φύλλου και ακόμα και στην περίπτωση της Inverse multiquadric που δεν εμφανίζονται αναστραμμένα τρίγωνα ίσως είναι ακατάλληλο για οποιαδήποτε προσομοίωση. Από την άλλη μεταξύ των Wendland C^0 και Wendland C^2 , η τελευταία επιτυγχάνει καλύτερα αποτελέσματα και μάλιστα δημιουργεί μια ζώνη γύρω από το φύλλο στην οποία η ποιότητα του πλέγματος παραμένει σχεδόν αμετάβλητη συγκριτικά με το αρχικό πλέγμα.

Όσον αφορά το Σχήμα 3 και σύμφωνα με τον Πίνακα 1 η υβριδική μέθοδος παράγει οριακά πιο ποιοτικό πλέγμα από την τυπική μέθοδο RBF. Αυτό που έχει ενδιαφέρον, σε αυτήν την περίπτωση, είναι ότι η ζώνη γύρω από το φύλλο στην οποία το πλέγμα διατηρεί την ποιότητα του είναι πιο παχιά όταν εφαρμόζεται η υβριδική μέθοδος. Κοντά στα στερεά όρια εμφανίζονται συνήθως οι μεγαλύτερες κλίσεις πίεσης και ταχύτητας και, συνεπώς, είναι αναγκαίο στις περιοχές αυτές το υπολογιστικό πλέγμα να είναι το δυνατόν καλύτερης ποιότητας.

Ακολούθως στον Πίνακα 2 παρουσιάζονται οι μέγιστες (κρίσιμες) γωνίες για τις οποίες το πλέγμα δεν εμφανίζει αναστραμμένα τρίγωνα μετά τη μετατόπιση του.

Συνάρτηση	ϕ_{crit}
Inverse quadratic	22°
Inverse multiquadric	33°
Wendland C^0 ($R_s = 11$)	48°
Wendland C^2 ($R_s = 11$)	69°
Wendland C^2 ($R_s = 11$) (hybrid)	62°

Πίνακας 2: Μέγιστη γωνία που μπορεί να στραφεί το φύλλο χωρίς να εμφανιστούν αναστραμμένα στοιχεία

Εφαρμογή στο πλέγμα γύρω από αεροτομή τεσσάρων στοιχείων

Σε αυτό το σημείο γίνεται εφαρμογή των προαναφερθεισών μεθόδων στο πλέγμα γύρω από αεροτομή τεσσάρων στοιχείων. Τα slats και τα flaps της αεροτομής στρέφονται

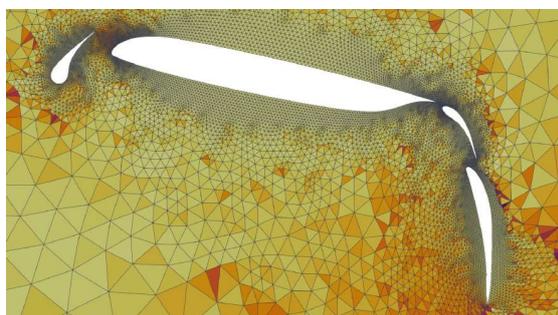
προς τα κάτω (όπως συμβαίνει κατά τη διάρκεια της προσγείωσης ενός αεροσκάφους) και κατόπιν το πλέγμα μετατοπίζεται κατάλληλα. Στον Πίνακα 3 παρουσιάζονται τα αποτελέσματα της μετρικής ποιότητας του πλέγματος για διάφορες συναρτήσεις ενεργοποίησης.

Συνάρτηση	\overline{f}_{ss}	$\sigma_{f_{ss}}$	$f_{ss,min}$
Exponential	0.933	0.066	0.106
Wendland C^0 ($R_s = 1$)	0.932	0.070	0.115
Wendland C^0 ($R_s = 5$)	0.934	0.067	0.108
Wendland C^2 ($R_s = 5$)	0.925	0.084	0.099

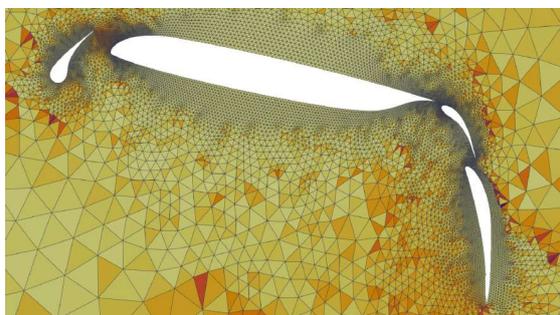
Πίνακας 3: Μέση τιμή, τυπική απόκλιση και ελάχιστη τιμή της μετρικής ποιότητας πλέγματος για διάφορες συναρτήσεις ενεργοποίησης.



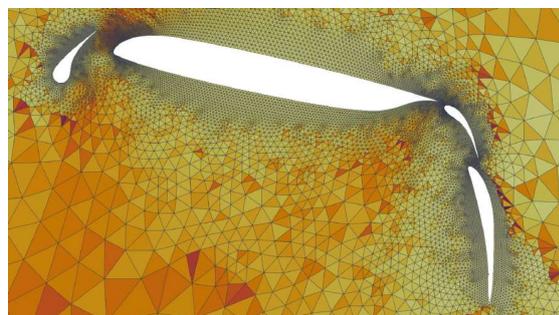
(α') *Exponential*



(β') *Wendland C^0 , $R_s = 1$*



(γ') *Wendland C^0 , $R_s = 5$*

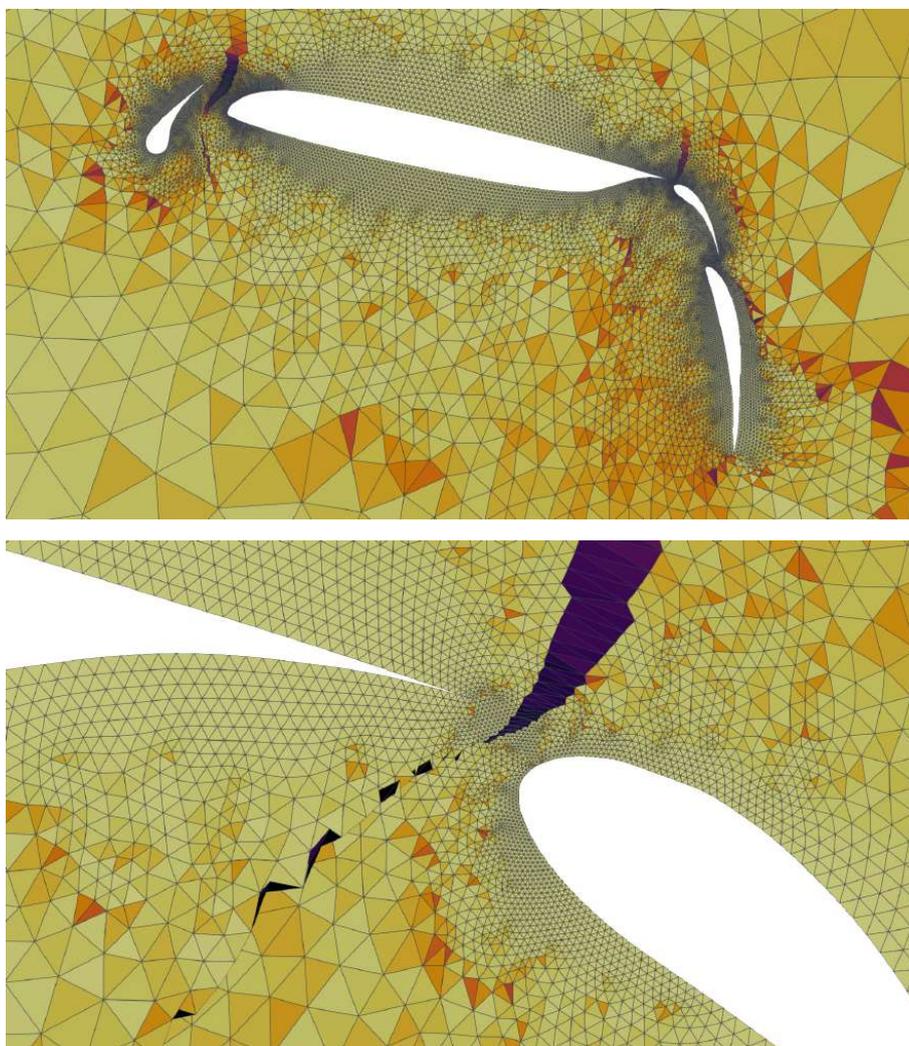


(δ') *Wendland C^2 , $R_s = 5$*

Σχήμα 4: Προκύπτοντα πλέγματα για μετατόπιση των *slats* και των *flaps* προς τα κάτω

Σύμφωνα με τον Πίνακα 3 όλες οι συναρτήσεις που δοκιμάστηκαν στην τυπική μέθοδο RBF δίνουν πλέγματα παρόμοιας ποιότητας. Αυτό είναι εμφανές και στο Σχήμα 4. Βέβαια παρατηρείται ότι καθώς η απόσταση από το όριο αυξάνεται η συνάρτηση Wendland C^0 και με τις δύο ακτίνες επιρροής που δοκιμάστηκε, παράγει στοιχεία ελαφρώς καλύτερης ποιότητας συγκριτικά με τις άλλες δύο συναρτήσεις ενεργοποίησης.

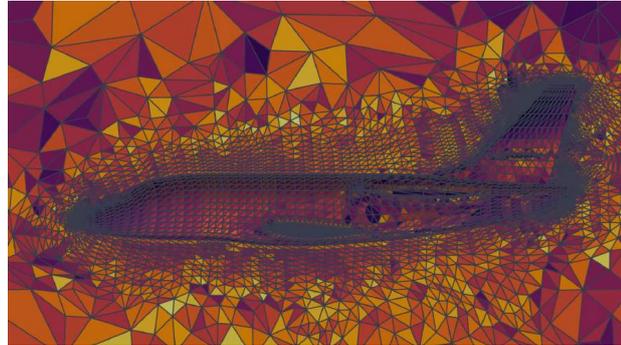
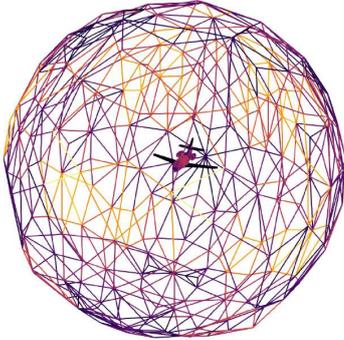
Ακολούθως, στο Σχήμα 5 εφαρμόζεται η υβριδική μέθοδος στο παρόν πλέγμα. Παρατηρείται η εμφάνιση αναστραμμένων τριγώνων μεταξύ στατικών και κινούμενων στοιχείων. Αυτό αποτελεί και ένα μειονέκτημα της υβριδικής μεθόδου. Όταν υπάρχει μεγάλη σχετική κίνηση ενός οριακού κόμβου ως προς κάποιον άλλο οριακό κόμβο, τότε οι εσωτερικοί κόμβοι του πλέγματος που βρίσκονται κοντά στη διάμεσο των δύο οριακών κόμβων μετατοπίζονται ασύμβατα μεταξύ τους και έτσι οδηγούν στην εμφάνιση αναστραμμένων στοιχείων στην περιοχή αυτή.



Σχήμα 5: Εφαρμογή της υβριδικής μεθόδου στο πλέγμα αεροτομής τεσσάρων στοιχείων. Αναστραμμένα τρίγωνα εμφανίζονται μεταξύ στατικών και κινούμενων στοιχείων.

Εφαρμογή στο πλέγμα γύρω από αεροσκάφος

Τέλος, γίνεται δοκιμή των μεθόδων στο τριδιάστατο πλέγμα γύρω από αεροσκάφος. Το αεροσκάφος στρέφεται γύρω από τον άξονα y (pitch axis) και κατόπιν το πλέγμα μετατοπίζεται κατάλληλα.



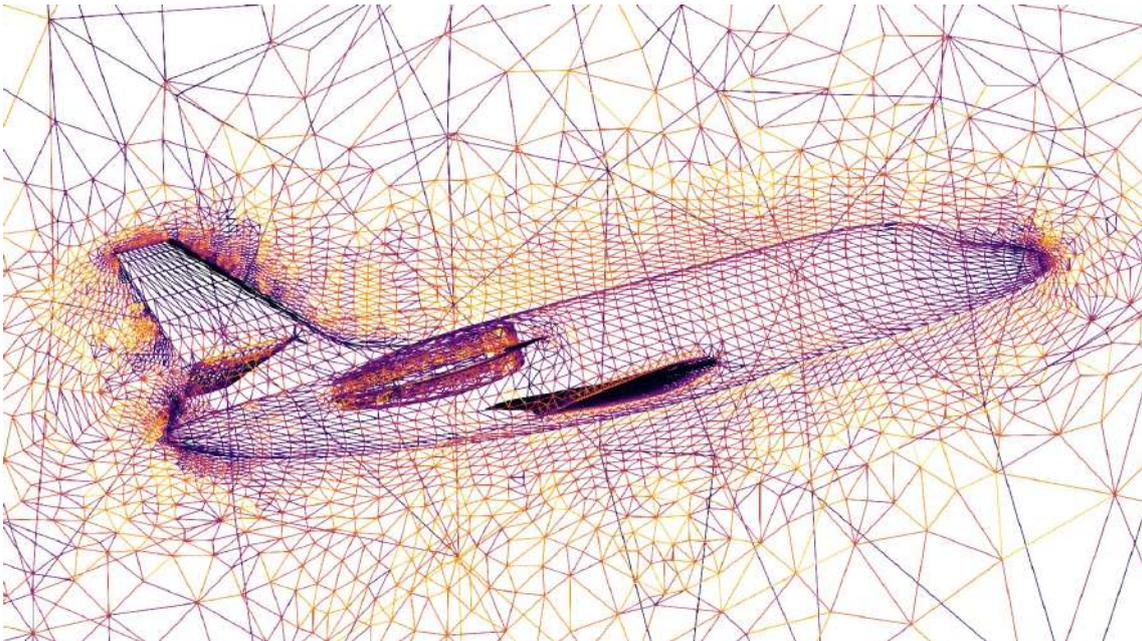
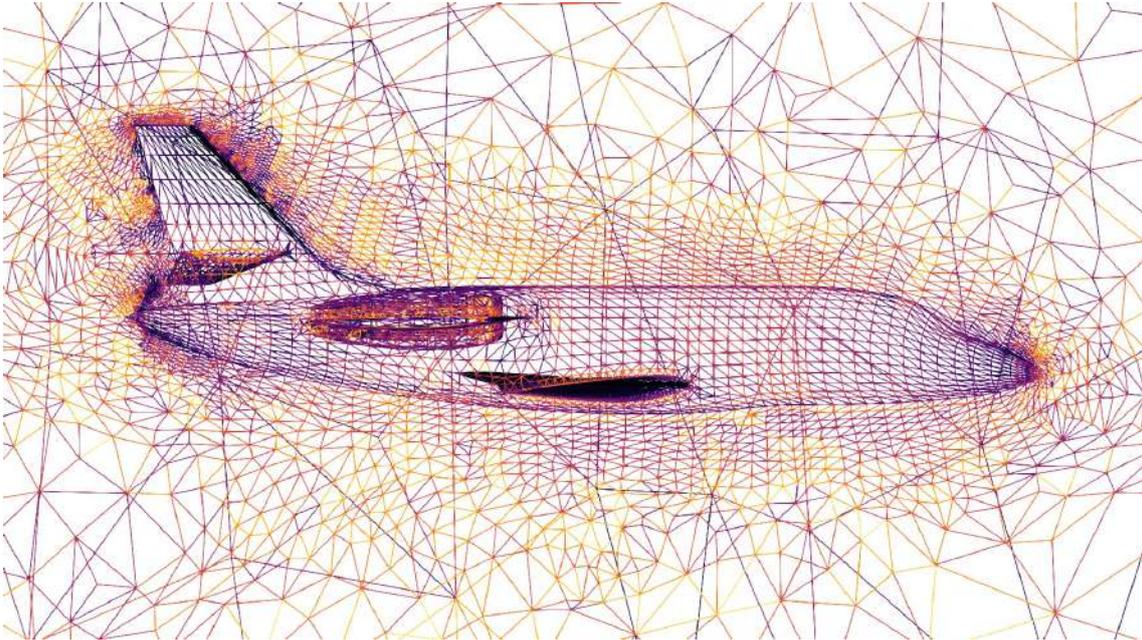
Σχήμα 6: Το αρχικό πλέγμα γύρω από το αεροσκάφος. Το πλέγμα αποτελείται από τετραεδρικά στοιχεία.

Στο συγκεκριμένο πλέγμα, λόγω του ότι αποτελείται από πολύ μεγάλο πλήθος στοιχείων (511888 τετράεδρα), οι μετρικές ποιότητας του δεν εξάγουν ιδιαίτερα χρήσιμα συμπεράσματα. Αυτό που εμφανίζει ενδιαφέρον είναι να δούμε σε ποια γωνία στροφής το πλέγμα εμφανίζει αναστραμμένα τετράεδρα. Το τελευταίο παρουσιάζεται στον Πίνακα 4

Συνάρτηση	ϕ_{crit}
Wendland C^0 ($R_s = 40$)	5°
Wendland C^2 ($R_s = 21$)	12°
Wendland C^2 ($R_s = 40$)	16°
Wendland C^2 ($R_s = 40$) (hybrid)	4°

Πίνακας 4: Μέγιστες γωνίες για τις οποίες στο πλέγμα δεν εμφανίζονται αναστραμμένα τετράεδρα

Παρατηρούμε ότι η τυπική μέθοδος RBF με συνάρτηση ενεργοποίησης τη Wendland C^2 είναι σαφώς αποδοτικότερη. Ειδικότερα όταν η τελευταία εφαρμόζεται με αρκετά μεγάλη ακτίνα επιρροής μπορεί να υποστηρίξει γωνία στροφής ίση με 16° . Στο Σχήμα 7 παρουσιάζεται το πλέγμα σε τομή μετά από στροφή 15° .



Σχήμα 7: Αρχικό και μετατοπισμένο πλέγμα για γωνία στροφής 15° . Η τυπική μέθοδος *RBF* εφαρμόζεται με συνάρτηση ενεργοποίησης τη *Wendland C^2* και ακτίνα επιρροής $R_s = 40$.

Βιβλιογραφία

- [1] Johnson, A. A. and Tezduyar, T. E.: *Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces*. Computer Methods in Applied Mechanics and Engineering, 119(1):73–94, Nov 1994, ISSN 0045-7825. 10.1016/0045-7825(94)00077-8.
- [2] Boer, A. de, Schoot, M. S. van der, and Bijl, H.: *Mesh deformation based on radial basis function interpolation*. Computers & Structures, 85(11):784–795, Jun 2007, ISSN 0045-7949. 10.1016/j.compstruc.2007.01.013.
- [3] Gagliardi, F. and Giannakoglou, K. C.: *A two-step radial basis function-based cfd mesh displacement tool*. Advances in Engineering Software, 128:86–97, Feb 2019, ISSN 0965-9978. 10.1016/j.advengsoft.2018.11.011.
- [4] Samareh, J.: *Application of quaternions for mesh deformation*. January 2002.
- [5] Maruyama, D., Bailly, D., and Carrier, G.: *High Quality Mesh Deformation Using Quaternions for Orthogonality Preservation*. Jan 2012. journalAbbreviation: 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition.
- [6] Selim, M. M. and Koomullil, R. P.: *Mesh deformation approaches – a survey*. Journal of Physical Mathematics, 7(2):1–9, Jan 2016, ISSN 2090-0899, 2090-0902. 10.4172/2090-0902.1000181.
- [7] Niu, J., Lei, J., and He, J.: *Radial basis function mesh deformation based on dynamic control points*. Aerospace Science and Technology, 64:122–132, May 2017, ISSN 1270-9638. 10.1016/j.ast.2017.01.022.
- [8] Strofylas, G. A., Lygidakis, G. N., and Nikolos, I. K.: *An agglomeration strategy for accelerating rbf-based mesh deformation*. Advances in Engineering Software, 107:13–37, May 2017, ISSN 0965-9978. 10.1016/j.advengsoft.2017.02.004.
- [9] Rendall, T. C. S. and Allen, C. B.: *Parallel efficient mesh motion using radial basis functions with application to multi-bladed rotors*. International Journal for Numerical Methods in Engineering, 81(1):89–105, 2010, ISSN 1097-0207. 10.1002/nme.2678.

- [10] Allen, C. B.: *Parallel universal approach to mesh motion and application to rotors in forward flight*. International Journal for Numerical Methods in Engineering, 69(10):2126–2149, 2007, ISSN 1097-0207. 10.1002/nme.1846.
- [11] Staten, M. L., Owen, S. J., Shontz, S. M., Salinger, A. G., and Coffey, T. S.: *A comparison of mesh morphing methods for 3d shape optimization*. In Quadros, William Roshan (editor): *Proceedings of the 20th International Meshing Roundtable*, page 293–311. Springer, 2012, ISBN 978-3-642-24734-7.
- [12] Batina, J. T.: *Unsteady euler airfoil solutions using unstructured dynamic meshes*. AIAA Journal, 28(8):1381–1388, Aug 1990, ISSN 0001-1452. 10.2514/3.25229.
- [13] Degand, C. and Farhat, C.: *A three-dimensional torsional spring analogy method for unstructured dynamic meshes*. Computers & Structures, 80(3):305–316, Feb 2002, ISSN 0045-7949. 10.1016/S0045-7949(02)00002-0.
- [14] Zeng, D. and Ethier, C. R.: *A semi-torsional spring analogy model for updating unstructured meshes in 3d moving domains*. Finite Elements in Analysis and Design, 41(11):1118–1139, Jun 2005, ISSN 0168-874X. 10.1016/j.finel.2005.01.003.
- [15] Bottasso, C. L., Detomi, D., and Serra, R.: *The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes*. Computer Methods in Applied Mechanics and Engineering, 194(39):4244–4264, Oct 2005, ISSN 0045-7825. 10.1016/j.cma.2004.08.014.
- [16] Huo, S. H., Wang, F. S., Yan, W. Z., and Yue, Z. F.: *Layered elastic solid method for the generation of unstructured dynamic mesh*. Finite Elements in Analysis and Design, 46(10):949–955, Oct 2010, ISSN 0168-874X. 10.1016/j.finel.2010.06.006.
- [17] Hsu, S. Y. and Chang, C. L.: *Mesh deformation based on fully stressed design: the method and 2-d examples*. International Journal for Numerical Methods in Engineering, 72(5):606–629, 2007, ISSN 1097-0207. 10.1002/nme.2027.
- [18] Stein, K., Tezduyar, T. E., and Benney, R.: *Automatic mesh update with the solid-extension mesh moving technique*. Computer Methods in Applied Mechanics and Engineering, 193(21):2019–2032, May 2004, ISSN 0045-7825. 10.1016/j.cma.2003.12.046.
- [19] Burg, C.: *Analytic study of 2d and 3d grid motion using modified laplacian*. International Journal for Numerical Methods in Fluids, 52(2):163–197, 2006, ISSN 1097-0363. 10.1002/fld.1173.
- [20] Helenbrook, B. T.: *Mesh deformation using the biharmonic operator*. International Journal for Numerical Methods in Engineering, 56(7):1007–1021, 2003, ISSN 1097-0207. 10.1002/nme.595.

- [21] Zhao, Y. and Forhad, A.: *A general method for simulation of fluid flows with moving and compliant boundaries on unstructured grids*. Computer Methods in Applied Mechanics and Engineering, 192(39):4439–4466, Sep 2003, ISSN 0045-7825. 10.1016/S0045-7825(03)00424-9.
- [22] Ghosal, S.: *On algebraic smoothing: Theory and results*. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 3, page 17–20 vol.3, Sep 2000.
- [23] Tsolovikos, A.: *Deformation of Computational Meshes Using Delaunay Graph Parameterization – Applications in the Adjoint-Based Aerodynamic Shape Optimization*. Diploma thesis, Lab. of Thermal Turbomachines, N.T.U.A., Athens, Jun 2018.
- [24] Witteveen, J.: *Explicit and Robust Inverse Distance Weighting Mesh Deformation for CFD*. American Institute of Aeronautics and Astronautics. <https://arc.aiaa.org/doi/abs/10.2514/6.2010-165>.
- [25] Thompson, J. F., Warsi, Z. U.A., and Mastin, C. W.: *Numerical Grid Generation: Foundations and Applications*. Elsevier North-Holland, Inc., USA, 1985, ISBN 044400985X.
- [26] Broomhead, D. and Lowe, D.: *Multivariable functional interpolation and adaptive networks*. Complex Syst., 2, 1988.
- [27] Broomhead, D. and Lowe, D.: *Radial basis functions, multi-variable functional interpolation and adaptive networks*. ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (UNITED KINGDOM), RSRE-MEMO-4148, Mar 1988.
- [28] Manton, J. and Amblard, P.: *A primer on reproducing kernel hilbert spaces*. Found. Trends Signal Process., 8:1–126, 2015.
- [29] Commons, Wikimedia: *Bed of nails interpolant*, 2019. https://en.wikipedia.org/wiki/File:Bed_of_nails.png, [Online; accessed August 09, 2021].
- [30] Fasshauer, G. E.: *Meshfree Approximation Methods with MATLAB*. World Scientific, 2007, ISBN 978-981-270-633-1. Google-Books-ID: gtqBdMEqyEC.
- [31] Wendland, H.: *Konstruktion und Untersuchung radialer Basisfunktionen mit kompaktem Träger*. 1996. <https://books.google.gr/books?id=4AG5HAAACAAJ>.
- [32] Quarteroni, A., Sacco, R., and Saleri, F.: *Iterative Methods for Solving Linear Systems*, page 123–181. Texts in Applied Mathematics. Springer, 2007, ISBN 978-0-387-22750-4. https://doi.org/10.1007/978-0-387-22750-4_4.
- [33] Greenbaum, A.: *11. Incomplete Decompositions*, pages 171–182. <https://epubs.siam.org/doi/abs/10.1137/1.9781611970937.ch11>.

- [34] Biancolini, M.: *Fast Radial Basis Functions for Engineering Applications*. January 2017, ISBN 978-3-319-75009-5.
- [35] Commons, Wikimedia: *An animation shows a plane get into gimbal lock*, 2019. https://commons.wikimedia.org/wiki/File:Gimbal_Lock_Plane.gif, [Online; accessed July 16, 2021].
- [36] Gumbert, C., Lohner, R., Parikh, P., and Pirzadeh, S.: *A package for unstructured grid generation and finite element flow solvers*. Jan 1989. <https://ntrs.nasa.gov/citations/19890060266>, NTRS Author Affiliations: NASA Langley Research Center, George Washington University, Vigyan, Inc. NTRS Report/Patent Number: AIAA PAPER 89-2175 NTRS Document ID: 19890060266 NTRS Research Center: Legacy CDMS (CDMS).
- [37] Ahrens, J., Geveci, B., and Law, C.: *Paraview: An end-user tool for large data visualization*. Visualization Handbook, Jan 2005.
- [38] Buhmann, M. D.: *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, Jul 2003, ISBN 978-1-139-43524-6. Google-Books-ID: TRMf53opzlsC.
- [39] Knupp, P. M.: *Algebraic mesh quality metrics for unstructured initial meshes*. Finite Elements in Analysis and Design, 39(3):217–241, Jan 2003, ISSN 0168-874X. 10.1016/S0168-874X(02)00070-7.